Information Brokerage via Location-Free Double Rulings [⋆]

Stefan Funke and Imran Rauf

Max-Planck-Institut für Informatik,
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
{funke,irauf}@mpi-inf.mpg.de

Abstract. The in-network aggregation and processing of information is what sets a sensor network apart from a pure data acquisition device. One way to model the exchange of information between the network nodes is to distinguish between nodes that are *producers* of information, i.e., 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. In this paper we aim to support that exchange of information via a so-called *information brokerage* scheme. Main features of our proposed scheme are that 1) it works in a location-free setting where nodes are unaware of their geographic locations 2) it is robust to non-regular network topologies and 3) it does not require the information producers and consumers to know of each other. Our proposed scheme employs boundary detection algorithms which only quite recently have been developed to extract geometry and topology information even in location-free network deployments.

1 Introduction

In their first generation, sensor networks were primarily considered a data acquisition device, where the data acquired at the sensor nodes was transferred to a central authority for evaluation and storage. Due to the rapid growth of the size of sensor network deployments, such a centralized model of operation becomes restricting as the amounts of data that can be transferred through the network is naturally limited. Recently, more emphasis is put on processing and interpreting the acquired data *within* the network. That is, sensor networks have made the step from a pure data acquisition device to a new form of computing device. For the *in-network processing* of sensor data, novel schemes for the information exchange between the sensor nodes are necessary which are more focused on the data itself rather than the identities of the individual network nodes. *Data-centric* processing, storage and retrieval of information has been the focus of several recent papers, see [10, 13, 8]. For example in GHT [13], each

---

[⋆] This work was supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC) funded by the German Federal Ministry of Education and Research (FKZ 01IMC01)

type σ_A of information (like occurrence of some event A) is mapped to a location using a *geographic hash function* ϕ, which is known to all nodes within the network. Upon occurrence of event A, a nearby sensor node detecting this occurrence sends a message to a node close to the location determined by $\phi(\sigma_A)$, indicating that is has gathered data of type σ_A. Any other node interested in information of type σ_A can obtain that information by sending a query message to $\phi(\sigma_A)$. Note that in this scheme, the network nodes do not have to be aware of their identities, i.e., a directory service as usually required for point-to-point routing is not necessary; the routes of the messages are purely determined by the *type of the data* (and the associated geographic locations). The presence of location information at the network nodes for this and several related approaches is fundamental, though.

In this paper we take a slightly different view on the information exchange between the nodes within a sensor network. We consider the network nodes as *producers* and *consumers* of information. A *producer* of information v_p, upon detection of some event A, sends messages along two paths within the network, leaving a trail indicating that v_p has gathered information of type σ_A[1]. A *consumer* of information v_c that is interested in retrieving information of some type σ_A sends messages along two paths querying for data of type σ_A. If one of the messages on its way visits a node which has information about type σ_A stored by some producer, it sends back a message to the consumer following the trail left by the consumer's message. What a *double ruling scheme* guarantees is that the trajectories of the messages of producer and consumer always meet each other. In Figure 1 we have depicted a very simple double ruling scheme where producers of information distribute their data on vertical paths and consumers look for information on horizontal paths. Observe that this simple scheme en-

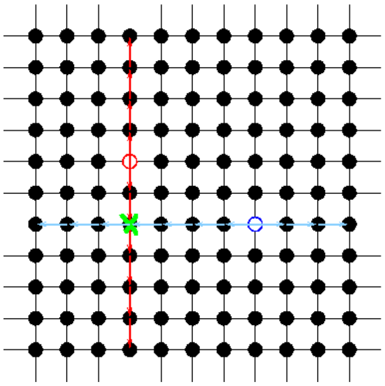

Fig. 1. Simplest double ruling scheme where informations producers and consumers are brought together at the node indicated by a green cross.

---

[1] The trail data expires after some time depending on the lifetime of the event itself.

joys a certain *distance-sensitivity*, that is, if a consumer is looking for an event that has happened nearby, the two trajectories of consumer and producer are guaranteed to meet quickly; distance-sensitivity is not provided by schemes like GHT ([13]): the location where data about a certain type of event is stored can be arbitrarily far away from both consumers and producers. This drawback is partly alleviated by the GLS [1] approach at a higher maintenance cost. All the aforementioned approaches require geographic location information at the network nodes, though.

The aim of this paper is to develop a information brokerage scheme that does not require location-information at the network nodes, yet it is based on the same simple intuition of the double ruling scheme in Figure 1. Due to the absence of location information, the directional information (vertical/horizontal) has to be replaced by topology-based gradient fields, and the property that the trajectories of information consumers and producers meet has to be ensured by a setup phase which extracts important topological properties of the network. The latter is achieved by the employment of boundary detection algorithms which only quite recently have been developed.

1.1 Related Work

The most prominent representatives of data-centric information storage and retrieval schemes are GHT (Geographic Hash Table, [13]) and GLS ([1]) as already sketched above. In very recent work, the concepts of geographic hashing and double rulings have been generalized in [11], where the authors propose a scheme in which both information dissemination as well as information retrieval is performed by sending messages along a closed replication curve. The respective curve is guaranteed to pass through a certain location as defined by a hash function (similar to GHT), still retrieval can successfully achieved in a distance-sensitive manner since retrieval and replication curve typically meet each other quite quickly. GHT, GLS, and the latter approach are crucially dependent on *geographic location information* at the network nodes. Fewer approaches have been proposed that also work in a location-free setting. Fang et al. in [2] combine the GLIDER routing scheme with a hashing approach on the top level similar to GHTs and local double rulings (within the topologically simple routable tiles). Since their approach is built on top of a routing scheme, though, it requires the permanent maintenance of rather complex global topology information throughout the network. Another approach, based on a hierarchical naming and routing scheme, was presented in [5]. It is essentially an extension of the GLS scheme ensuring distance-sensitivity in location-free scenarios if the underlying routing scheme produces close-to-optimal routes. Again, being based upon a point-to-point routing scheme, a considerable amount of global topology information has to be constantly maintained throughout the network. Unfortunately, the simple approach of vertical and horizontal message trajectories as sketched in Figure 1 does not easily extend to more complex network topologies (i.e. non-grid-like, potentially with holes).

1.2 Our Contribution

We develop an information brokerage scheme that is based on the same simple intuition of the double ruling scheme in Figure 1. Main features of our scheme are that it is *location-free*, i.e. does not rely on geographic location information at the network nodes (like [13, 1, 11] do). Our scheme is *distance-sensitive*, i.e. the cost for retrieving information about some event is proportional to the distance to where the event happened (unlike [13, 1]), and it is not built on top of a point-to-point routing scheme (like [2, 5]) and hence does not require the maintenance of extensive global topology information during operation. Instead, after a setup phase, the only information that needs to be maintained are two scalar values at each network node corresponding to the two directions of the double ruling scheme. Changes in the network topology can be easily dealt with by repeatedly checking and updating the scalar values based on the values stored with communication neighbors. Due to the absence of location information, the directional information (vertical/horizontal) as used in Figure 1 has to be replaced by topology-based gradient fields, and the property that the trajectories of information consumers and producers meet has to be ensured by a setup phase which extracts important topological properties of the network. We also evaluate our information brokerage scheme in simulations and show that it behaves favorably compared to for example GHT or simple location-based double ruling schemes. 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 are mainly interested in allowing consumers to quickly obtain the desired information while keeping the cost of information diffusion for the producer reasonable.

2 Location-Free Double Rulings

2.1 Intuition in the Continuous Case

To explain our approach we first consider a continuous (not necessarily simply connected) domain where all the sensors are deployed; see Figure 2, left, for an illustration (the holes in the domain represent communication voids as for example induced by obstacles like buildings). The idea of our double ruling scheme is to choose one of the boundary cycles of the domain and partition it into four pieces G_-, G_+, H_-, H_+ such that the G and H pieces alternate, see Figure 2, center. A producer of information upon detection of an interesting event, sends a message to both G_- and G_+, essentially simulating the vertical paths in the simple scheme from Figure 1. A node searching for information, on the other hand, sends a message two messages to H_- and H_+ (corresponding to the horizontal paths); see Figure 2, right.

Several interesting observations are worth noting: 1) It does not matter which boundary cycle (outer boundary cycle or boundary cycle of a hole) we choose, if we partition it in the above described way, the paths of producers and consumers

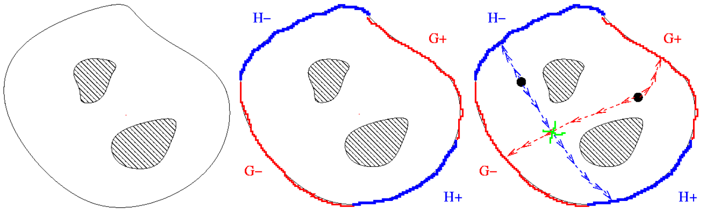

Fig. 2. Intuition of our approach in the continuous setting.

are guaranteed to cross. For reasons of load-balancing, though, it is preferable to partition the largest boundary cycle which typically is the outer boundary cycle; partitioning of a small cycle leads to a lot of producers' or consumers' paths that have to go through a small part of the network. 2) It would also be possible – instead of partitioning a boundary cycle into 4 pieces – to simply choose 4 *points* on the boundary cycle, and defining the producers' or consumers' paths towards those single points. But again, this would lead to a high load imbalance within the network. 3) For correctness it is irrelevant on which paths the messages reach G_-/G_+ (H_-/H_+ respectively); to limit the overall traffic load it is generally advisable to prefer relatively short paths, though.

2.2 Translation to the Discrete Setting

Translating the idea from the continuous setting to real wireless sensor network deployments incurs two major difficulties. First of all, in a real network deployment is only a discrete sampling of the continuous domain. Secondly, the above description heavily relies on the *geometry* of the domain. But in many application scenarios the network nodes only know approximately about their geographic positions or not at all; equipping every single node with a GPS receiver is typically not feasible due to cost reasons, in particular for a large number of deployed nodes. To overcome these difficulties we will employ algorithms and techniques that have recently been developed under the topic of *boundary detection*. Here one assumes that the communication graph of a wireless network resembles a *(quasi)-unit-disk graph* and the goal is to identify nodes close to network boundaries just by inspection of the communication graph without using any geometric location information. Fortunately the communication graph of a wireless network implicitly contains enough information such that this is possible as has been shown in a sequence of papers [4, 3, 6, 15]. See Figure 3 for a sketch of the output of the boundary detection algorithm in [6]; note that the algorithm actually produces connected sets of nodes which form the boundaries, the subsampling as a set of few boundary nodes was only chosen for visualization purposes.

The intersection of producer's and consumer's paths in the discrete setting either happens at a common node on these paths, or an edge from one path crosses an edge of the other. Since one of the nodes in a pair of crossing edges must have edges to all three other nodes, the successful rendezvous of consumer's

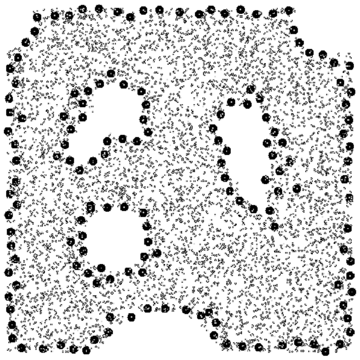

Fig. 3. Output of a boundary detection algorithm.

and producer's paths is guaranteed by replicating the message to the 1-hop neighbors of the nodes on these paths.

The main steps of our approach are as follows:

1. use a boundary detection algorithm to identify the outer boundary of the network (we use the algorithm proposed in [6])
2. partition the outer boundary into 4 well-behaved pieces (see later for potential caveats)
3. construct gradient-fields between opposing boundary pieces for the double ruling scheme

In the following we go into a bit more detail regarding each of these steps.

Identification of the outer boundary cycle As we noted earlier, any boundary cycle is sufficient to implement our double-ruling scheme but for better load-balancing we prefer an outer boundary cycle. Note that in the discrete setting, it is impossible to identify exactly the true geometric boundary without the location information of nodes at hand. Fortunately, boundary-detection algorithms can be employed to identify nodes which are 'near' the boundary. For example, the boundary detection algorithm by Funke et al. [6] guarantees to mark a node close to every point on the true geometric boundary.

To determine an outer cycle, we first use the algorithm in [6] to mark nodes near all boundaries. The marked nodes would also contain nodes near hole boundaries, however it is reasonable to assume that the largest connected component B of those marked nodes constitutes the outer boundary. Our algorithm then starts with a cycle in B and tries to 'grow' it until all other network nodes are connected in one component when removing B including its 1-hop neighborhood. To begin with, we arbitrarily pick some node u in B. Let $v \in B$ be the node at maximum distance from u, p_1 be the shortest path from u to v in B, and $w \in p_1$ be the node halfway between u and v on p_1. We mark w along with nodes that are up to 4-hop away from it. This essentially cut p_1 in the middle (see [6] for a more detailed description of this procedure), the alternative path p_2 from u to v combined with p_1 yields the initial cycle C. For the next phase, let $\mathcal{S}$ be the set of connected components after removing C together with its 1-hop neighbors and $S \in \mathcal{S}$ be the largest of these connected components. We try to grow S as much as possible by introducing more nodes to it and updating the outer cycle as needed. Every node u which is in the 1-hop neighborhood of S (and C as well) tries to include itself into S. Let $F \subseteq C$ be the set of neighbors of u in C, L be the largest connected component of $C \setminus F$, and l_1 and l_2 be the extreme points of L. If there is l_1-l_2 path p which do not pass through nodes in $S \cup F$ then we replace the current cycle with the one formed by L and p. We keep updating C in this manner until no more node can be added to S.

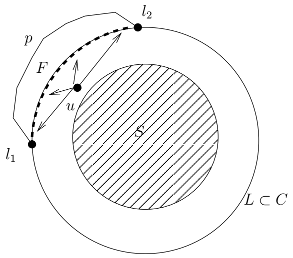

Fig. 4. Growing C by updating the current cycle with the one formed by L and p.

Partitioning of the outer boundary cycle An important aspect that influences the load-balancing property of the producers' and consumers' paths is the partitioning of the outer boundary cycle. For a badly chosen partition these paths may concentrate in certain regions and thus cause congestion in the network. Figure 5, left shows the temperature gradient field (as described in the following) when the partition of outer-cycle of square-shaped network is taken to be the sides of the square, while Figure 5, right considers the other case when the partitions extend across the corners of the square. Note that the contours in the former case are uniform while in later case they are more concentrated at corners which will lead to higher load on those nodes. This observation suggests to select the partition that somehow respect the geometry of the network to have better load-balancing (without using the geographic location information, though).

Our heuristic tries to identify convex corners in the network boundary and cut the cycle at these corners. For a node x on the outer cycle C, let $u, v \in C$ be two nodes which are k distance apart from x on C in the opposite directions, where $k \leq |C|$. Consider the ratio $r(x) = \frac{2k}{\mathrm{dist}(u,v)}$, where $\mathrm{dist}(u, v)$ represents the hop count distance between u and v in the communication graph. Note that if x is near a convex corner of the sensor network, the distance $\mathrm{dist}(u, v)$ is much smaller than $2k$. This observation suggest to select four corners (cut-points)

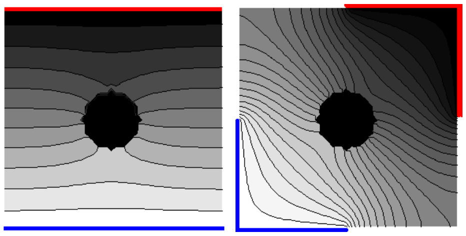

Fig. 5. Gradient field for square-shaped network and different partitions.

which are sufficiently far apart and have highest $r(x)$ value. Using $k = |C|/8$, we pick them one by one and selecting the next which is at least $k + 1$ distance from previously selected corners and has the highest ratio.

Construction of temperature gradient-fields for the double ruling Given a partition of outer cycle G_-, G_+, H_-, H_+ such that the G and H pieces alternate, a straight forward way to define producers' and consumers' paths is to use the hop count towards these pieces. More precisely the producer announces its message on shortest paths towards G_- and G_+, whereas the consumer look for the desired information by sending a message on shortest paths towards H_- and H_+. The required hop counts (distances to the respective pieces) can be computed by 4 breadth-first searches each from a piece of the partition of outer cycle C.

The problem with this approach is that it is very sensitive to the structure of the pieces of the outer cycle. For example when let's say piece G_- has a protruding vertex v_-, most traffic sent towards G_- will actually be sent towards v_- leading to high load imbalance. Another idea which is inspired by Skraba et el [14] is to compute potential functions g and h on nodes whose gradient define the producers' and consumers' paths respectively, and which are not as sensible to protruding vertices. In [14], the gradient field g is set up by fixing the temperature of nodes in G_- to 0 and to 1 for nodes in G_+, and iteratively setting the temperature of the other nodes to average their neighbors until conversion (analogously for H_+/H_-). This process effectively solves a Laplace's equation with Dirichlet boundary conditions over the network. As noted in [14], iteratively solving Laplace equation discretized on N grid-points is well-studied problem and it takes $\mathcal{O}(N)$ time in 2D [16]. They also observed similar running times for graphs with constant degree as usually encountered in wireless networks. To speed-up the convergence, we set the initial values of g and h to be the relative distance to G_+ and H_+ respectively before starting the iterative process as it was already suggested in [9] to speed up the computation of a similar function.

While the initial computation requires some effort, we want to emphasize that in terms of adaptation to dynamic changes in the connectivity structure of the network, this approach of iteratively averaging requires almost no effort apart from periodically checking the temperatures of the neighbors. Small changes in the network topology are also not expected to change the gradient field drastically, so stabilization can be achieved quite quickly.

3 Simulations

To evaluate the performance of the algorithms described above, we performed a set of computer-simulated experiments considering four variants of our approach, distinguishing between a hop-count-based gradient field and a temperature-based gradient field (**Hop/Temp**) as well as shape aware or unaware outer boundary partitioning (**SHA/SHU**). Main objective was to analyze the differences in terms of load balancing, but we also compared our best variant (Temp/SHA) with GHT in terms of responsiveness for the consumers (note, though, that the routing protocol GPSR [7] which GHT is based upon requires geographic location information which our approach does not). The Gabriel graph was used for face routing in GPSR.

Our simulator [12] is not packet-based, and thus it does not take into account some issues that occur in practice (i.e. medium access and message loss). However, we feel that these factors would have similar impact on all algorithms, and thus would not significantly affect the relative performance.

3.1 Load balancing

Let us first consider the amount of load (number of packets) received by nodes during information exchange between producers and consumers. We compare the distribution of load on nodes for different variants of our scheme.

Load		250	450	650	850	1050	1250	1450	1650	1850
Inverse Cumulative Frequency	Hop/SHA	826	366	192	104	61	25	10	5	2
	Temp/SHA	694	95	16	4	2	2	0	0	0
	Temp/SHU	605	190	93	30	23	17	14	4	3
	Hop/SHU	626	268	138	90	69	48	33	22	17

Table 1. Inverse cumulative frequencies of load for various double ruling schemes. (a) Hop-Count/Shape aware, (b) Temperature/Shape aware, (c) Temperature/Shape unaware (d) Hop-Count/Shape unaware

The simulations in this section are performed on a network containing 12000 nodes distributed uniformly at random with average degree of 29. Every node in the network is both producer and consumer of information of single data type. This will essentially flood the network where every node is interested in getting information from every other node. Table 1 shows the inverse cumulative frequencies of nodes i.e. number of nodes having load greater than certain values. Figure 6 presents the same as plot. Our scheme clearly outperform others by exhibiting fewer nodes with high load.

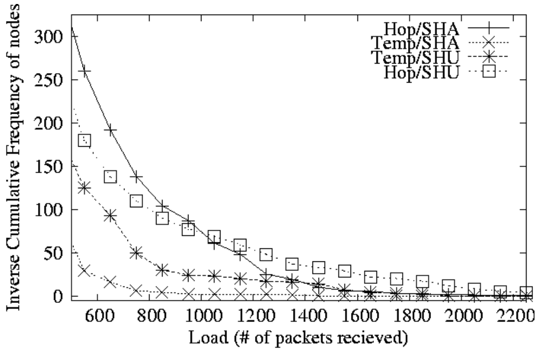

Fig. 6. Plot of Table 1.

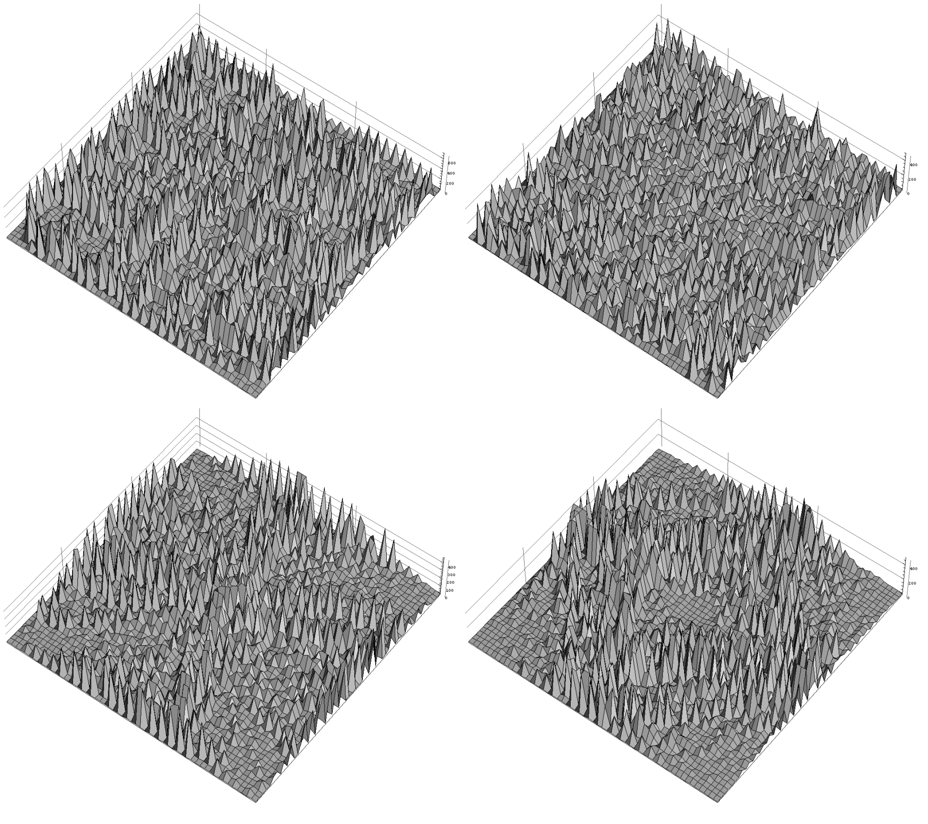

Fig. 7. Load distribution for various simulations presented in Table 1: Hop/SHA (top-left), Temp/SHA (top-right), Temp/SHU (bottom-left) & Hop/SHU (bottom-right).

Figure 7 shows the actual load distribution in the network during our simulations of Figure 6. We observe that, using the hop count for the double ruling gives nodes with high loads scattered all over the network, while choosing a bad boundary partition yields high loads near the breakpoints between G_-, G_+, H_-, H_+. This is due to many paths being attracted towards these corners. The scheme with bad boundary partition and shortest path gives the worst results in all of the schemes considered. Employing both, a temperature-based gradient field as well as shape aware partitioning of the network delivers the most balanced load distribution with only few exceptions near the corners.

3.2 Load Balance for Non-regular Network Topologies

Let us now examine how robust our approach is in the presence of holes or communication voids in the network. Table 2 shows the simulation results on a square network containing single hole in the middle, while Figure 8 shows the same as plot. The schemes using shortest paths to define producers' and consumers' paths exhibit rather high load since shortest paths tend to 'hug' the hole boundaries quite closely, the temperature based variant with shape aware boundary partition again performs best.

Load		250	450	650	850	1050	1250	1450	1650	1850
Inverse Cumulative Frequency	Hop/SHA	608	273	153	113	71	42	22	13	5
	Temp/SHA	538	82	17	8	5	4	2	0	0
	Temp/SHU	461	180	95	41	20	11	8	2	1
	Hop/SHU	475	208	114	84	57	28	18	15	15

Table 2. Inverse cumulative frequencies of load for various double ruling schemes on a network with a hole in the middle (a) Hop-Count/Shape aware, (b) Temperature/Shape aware, (c) Temperature/Shape unaware (d) Hop-Count/Shape unaware

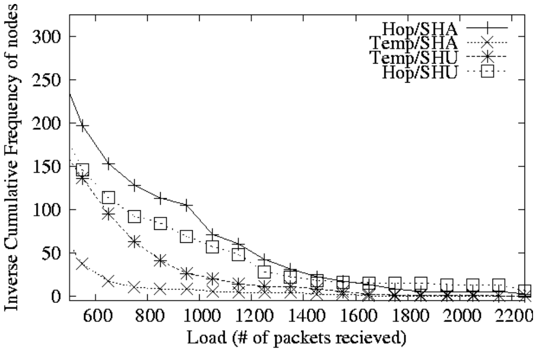

Fig. 8. Plot of Table 2.

Also for comparison we consider the simple geometric double ruling scheme, where producers (consumers) simply send their messages along horizontal (vertical) paths. Although this simple scheme gives good results for rectangular (axis-parallel) shaped networks, our scheme (Temp/SHA) exhibits better load balancing properties for networks with holes and more complicated shapes. Figure 9 and 10 gives two example of such cases.

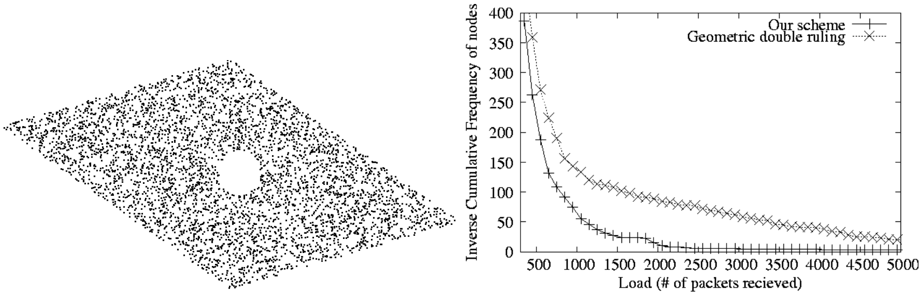

Fig. 9. Comparison with simple geometric double ruling.

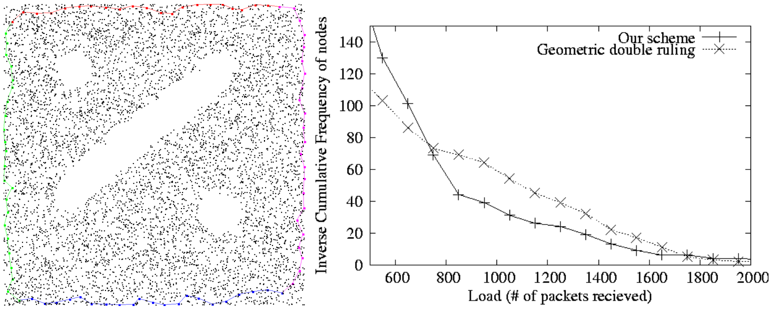

Fig. 10. Comparison with simple geometric double ruling.

3.3 Distance sensitivity

Our double ruling scheme has the property that the distance consumer needs to travel to get information from a producer is proportional to the actual distance between them. While in GHTs, a consumer need to travel to hashed location to get information even when the actual distance between producer and consumers is relatively short. We simulated both schemes and consider the average hop-distance a consumer needs to travel for varying distances between producer and consumer. In a network of 10000 nodes distributed uniformly at random, we define 1000 data terms and randomly pick a pair of producer and consumer for each of the type. Figure 11 presents the simulation results. Our scheme exhibits a direct relation between the actual distance of producer and consumer versus the cost incurred by a consumer to get the information, while the GHTs shows no such relation (since GHTs hash to a random location in the network which is typically quite far away).

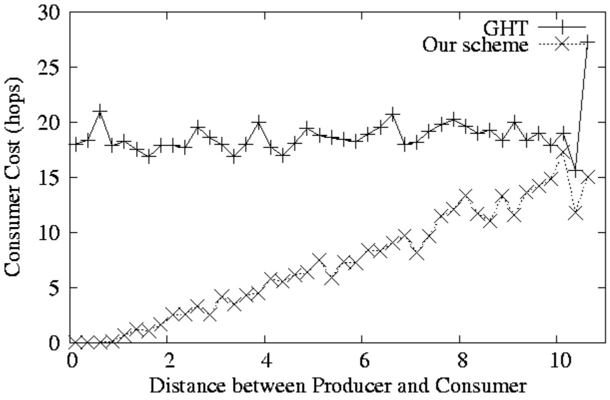

Fig. 11. Average hop-distance consumer need to travel for varying distances between producer and consumer.

4 Conclusions

In this paper we have presented a location-free double ruling scheme for information brokerage within a wireless sensor network. The absence of geographic location information requires some novel ideas how to carry over the simple intuition of the basic geometric double ruling scheme. The resulting approach exhibits quite good load balancing properties and also performs favorably in terms of distance-sensitivity from the consumer's point of view when for example compared with popular schemes like GHTs. On the downside, our approach – since based on a boundary recognition phase and the computation of the respective gradient fields – requires some time for startup until it can fully operate. Once in operation, it can rather easily deal with network volatility, though. Directions for future research include simplification of the startup phase as well as a more thorough investigation of the properties of the temperature gradient field to further decrease load imbalance.

References

1. 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 Commu-*

nication Society (INFOCOM), 2005.

2. Q. Fang, J. Gao, and L. Guibas. Landmark-based information brokerage in wireless sensor networks. In *Proc. 25th Conference of the IEEE Communications Society (InfoCom)*, Barcelona, Spain, April 2006.
3. S. P. Fekete, A. Kröller, D. Pfisterer, and S. Fischer. Deterministic boundary recognition and topology extraction for larg e sensor networks. In *Proc. of SODA*, 2006.
4. S. Funke. Topological hole detection in wireless sensor networks and its applications. In *DIALM-POMC '05: Proceedings of the 2005 joint workshop on Foundations of mobile computing*, pages 44–53, New York, NY, USA, 2005. ACM Press.
5. S. Funke, L. J. Guibas, A. Nguyen, and Y. Wang. Distance-sensitive routing and information brokerage in sensor networks. In *IEEE International Conference on Distributed Computing in Sensor System (DCOSS'06)*, pages 234–251, June 2006.
6. S. Funke and C. Klein. Hole detection or: "how much geometry hides in connectivity?". In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*, pages 377–385, New York, NY, USA, 2006. ACM Press.
7. B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM Press.
8. 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.
9. 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.
10. 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.
11. R. Sarkar, X. Zhu, and J. Gao. Double rulings for information brokerage in sensor networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 286–297, New York, NY, USA, 2006. ACM Press.
12. D. Schmitt. WNETS - a framework for testing and evaluating algorithms and models for wireless sensor networks. Master thesis, Universität des Saarlandes, 2007.
13. 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.
14. P. Skraba, Q. Fang, A. Nguyen, and L. Guibas. Sweeps over wireless sensor networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 143–151, New York, NY, USA, 2006. ACM Press.
15. Y. Wang, J. Gao, and J. S. B. Mitchell. Boundary recognition in sensor networks by topological methods. In *Proc. of the ACM/IEEE International Conference on Mobile Computin g and Networking (MobiCom)*, September 2006.
16. D. Young. *Iterative Methods for Solving Partial Difference Equations of Elliptic Type*. PhD thesis, Harvard University, 1950.