2011 IFIP Ninth International Conference on Embedded and Ubiquitous Computing

# Greedy Routing on Hierarchical Clusters

Ghazi Bouabene, Manolis Sifalakis and Christian Tschudin
Computer Science & Mathematics Department
University of Basel, Switzerland
Email: {ghazi.bouabene, sifalakis.manos, christian.tschudin}@unibas.ch

***Abstract*—Greedy geometric routing is one of the most appealing options for sensor and ad-hoc networks due to its simplicity, scalability and mainly stateless operation. However, its performance may vary when a low-dimensional embedding results in a graph with void regions and local minima locations. In this paper, we advocate the benefits of *clustering* for improving the performance of greedy routing over an existing embedding. Our contributions include: (a) an examination of the effects of clustering on influencing routing near void regions and overcoming problems with local minima; (b) a generic framework for enabling a cluster view for greedy routes, that can work with any low-dimensional embedding; and (c) a quantitative evaluation of the performance improvement we achieve, using some reference clustering approaches, over a number of virtual coordinate systems for geometric routing.**

***Index Terms*—Greedy routing, Clustering, Graph embedding, Sensor networks**

## I. Introduction

After more than a decade of research on routing for sensor and ad-hoc networks, geometric routing seems to be one of the most attractive options. Compared to their stateful counterparts, geometric routing protocols entail a low route-discovery overhead and are robust to topology changes. Most importantly, nodes do not need to maintain destination routing information and forward packet based only on local state. These features sum up to better scalability, energy conservation and increased robustness in face of volatile network conditions.

In geometric routing a rather simple stateless distributed algorithm routes messages from node to node along a source-destination path, based only on position information (node coordinates). A greedy routing algorithm forwards every packet to the node in the one-hop neighbourhood, which lies closest to the final destination.

In order to enable greedy routing, however, node coordinates need to reflect the relative network distances between nodes. As the assignment of node coordinates might not be a perfect approximation of network distances or due to the presence of obstacles, in practice, simple greedy routing is not always guaranteed to work. Depending on the geometry of the resulting network graph, the density of the network and the presence of void regions in the graph, the forwarding process might halt at local minima nodes with no neighbour closer to the destination than themselves.

In this paper we explore the effects of clustering as a low cost approach for improving the performance of greedy routing. The key idea is to re-use the connectivity information acquired during the pre-processing phase, and construct a graph of interconnected clusters that form a reduced connectivity topology. Node-local routing decisions thereafter, by means of the same greedy algorithm, can be performed either at the *base level* (based on actual node coordinates) or at the cluster level (based on cluster coordinates). The assumption we make and empirically evaluate in this work is that local minima and void regions in the connectivity graph of a sensor field (base level) are likely to be transposed or reformed at the cluster level view, which when combined with routing at the base level graph are often overcome. Our aim is therefore to measure to which extend such additional low-cost embeddings would improve the greedy routing performance compared to a single high-dimensional one.

The remaining of this paper is organised as follows. In Section II we describe the evolution of the field of greedy routing and briefly report on some of the major advancements to date. Section III provides a description of our framework for deploying clustering and constructing hierarchical cluster views (levels) for greedy routing. In Section IV we provide an evaluation over multiple configurations, of clustering methods, coordinate embeddings, and topologies, reporting a significant improvement in the effectiveness of greedy routing. These results essentially ratify our vote for simplicity. Section V provides some general insightful conclusions on our observations, and finally section VI summarises and concludes the paper.

## II. Related work

Early work on geometric routing was focused on actual geographic coordinates (e.g. [1], [2], [3] and other), which fuelled research on localisation and approximation of the actual network node coordinates from partial geo-positioning information [4], [5], [6], [7]. As node coordinates involved geo-positioning, addressing routing problems related to the morphology of the coordinate space (presence of void areas) led to hybrid routing approaches such as face routing [1], [8], [9] and other [10], [11]. Most of them however, introduce constraints and algorithmic complexity [12], [13], [14], without solving the routing problem in a real world deployment [15].

While geographic positioning coordinates provide a natural metric space for geometric routing, *virtual coordinates* offer an alternative whereby the coordinates assigned to nodes are not implied from physical locations but can be *chosen* to fit certain criteria. Such criteria typically include the shape and dimensionality of the coordinate space, the convexity of void regions, and other, which aim to enhance the performance of greedy routing itself without resorting to face routing.

978-0-7695-4552-3/11 $26.00 © 2011 IEEE
DOI 10.1109/EUC.2011.40

IEEE computer society

The task of assigning virtual coordinates to network nodes can be generalized as the problem of graph and metric space embedding [16], [17]: I.e. of finding a set of points in a metric space such that the distances between them accurately reflect the distances between the network nodes in a graph. Two popular methods to solve this problem explored in the networking literature are the Lipschitz embeddings [18] and multi-dimensional scaling techniques [16].

In the cases of Lipschitz embeddings [19], [7], a number of nodes are chosen as landmarks and each network node's coordinates are then expressed as the vector of distances (graph distance, network delay, etc) from all or a reduced set of landmarks.

In multi-dimensional scaling (MDS) [20], [21], [4] the focus is on reducing an error measure between the distances in the embedding space and the respective node distances in the connectivity graph. In the approaches that use classical MDS [16], [4], network nodes are first embedded in high dimensional space and then projected on a desired low-dimensional subspace that minimizes an error function called *strain*. This is the squared difference between the coordinates inner-product matrix and the double centered network distances one. In the more generalised forms of metric MDS the minimized error is Kruskal's *stress 1* [22] that is the normalized sum of squared differences between distances in the network and those in the embedding space. The error minimization can be computed centrally using non-linear programming, or distributed by simulating force-based systems [20], [23]. In such systems, pulling forces acting between the nodes are inversely proportional to their network distances. Force based systems usually find local minima that are "worse" than those found by centralized methods, however, the results are generally satisfactory given the significant trade-off of lending to simple distributed computation, and using only local information.

The use of solely virtual coordinates for greedy geometric routing was first proposed in [20]. Due to its elegant simplicity and good performance, this seminal work has been used thereafter as a reference system for comparisons. At the same time it also fuelled further research on understanding under what conditions (topology of the graph, adjustments of the routing metric) low-dimensional graph embeddings on the plane guarantee successful greedy routing [17], [24], [25]. A general insight in these works is that not every network graph admits to a Euclidean embedding that guarantees greedy routing. On the other hand, a hyberbolic embedding does [26], [27] and by extension, a sufficiently "morphed" planar embedding may also guarantee greedy routes [28]. While acknowledging the significance and theoretical foundations of these contributions, in this paper we chose to empirically explore if simple and computationally more sustainable tactics that lend to immediate deployment with small sensors, can yield comparable effects (i.e. overcome local minima, nullify void regions). As we show later, our first results support this intuition.

Finally a work, which bears a notable relevance to the one presented here, is presented in [10]. The authors exploit a clustering method to divide a network in compartments (*tiles*), based on which they factor the routing process in inter- and subsequently intra-compartment routing. The clustering method creates a *Voronoi complex* around a set of appointed landmarks (cluster heads), which then function as "attractors" for the different segments of routing paths. A cluster graph is the result of a *Delauney triangulation* among the landmarks.

Our work in this paper goes beyond a specific cluster mechanism and is more broadly scoped on the effects of clustering. From this end, we observed that it is actually more effective to start routing at the base level and resort to the cluster level only when local minima are found, by contrast to the strategy in [10] where routing always starts at the tile-level. Another realisation from our exploration is the un-necessitated need for landmarks to achieve the path-curving effects of [10]. Finally we do not resort to explicit planarisation of the embedded coordinate space.

## III. A CLUSTER-BASED APPROACH

To test our ideas and hypotheses we developed a simple framework that enabled us to experiment with different clustering algorithms and embedding approaches in Matlab. The main components of this framework, which are detailed in the following sections are responsible for the creation of clusters, their embedding in a virtual coordinate space, the management of routing state, and finally for applying a greedy routing strategy based on the selected distance metric.

### A. Construction of a cluster graph

Our method requires a cluster graph view of the network. Most of the proposed virtual coordinate schemes rely on a global preprocessing phase to discover the network connectivity, and thereafter a local routing phase where a greedy algorithm is used to forward messages across the network. During the initial phase a cluster graph can be produced by assigning the network graph nodes to groups (clusters) and then consider the adjacency relations between the different formed clusters.

One can use any of the graph clustering techniques available in literature [29], such as spectral clustering, minimum-cut based approaches, agglomerative methods, or other. However, a key requirement for efficiency in a wireless sensor network is that the clustering algorithm is simple enough for fast and distributed computation. To this end, clustering protocols from the wireless sensor networks literature such as [30], [31] can be applied. Naturally the quality of a clustering method in place, will affect the intra-cluster density as well as the cluster graph connectivity, and therefore is likely to influence the performance of greedy routing. As our goal is to study the average qualitative effects of clustering, rather than to seek a highly effective clustering algorithm, we experiment with two very different algorithms (of diverse sophistication) and multiple seeded configurations. This enabled us to produce a broad variety of cluster graphs for the evaluation. The algorithms we used were a spectral clustering variant of [32], [33], and at the other end of the range we employed a simple agglomerative approach, both of which are discussed further in the evaluation section.

### B. Cluster graph embedding

In the second step we need to embed the cluster graph so as to be able to perform greedy routing on it. It is possible to produce a new embedding into any metric space (not necessarily Euclidean) and of any dimensionality. However proposing a new embedding is beyond our scope as it would bias our objective assessment of the effects of clustering over existing systems. For this reason, we apply the same embedding technique that is being used at the base level. At the same time, as we discuss later on, this enables multi-level scaling and is aligned with our second goal to keep the deployment cost minimal by re-using the existing algorithmic code-base.

Our reference systems have been NoGeo, classical MDS and metric MDS and therefore we resort to the same embedding techniques to produce the embedding of the cluster graph in a $d$-dimensional Euclidean space ($d$ being equal to the dimensionality of the base graph embedding). For more information on these systems we refer the reader to [20], [16]. At the end of this process each cluster receives a coordinate in a metric space, such that the same greedy routing algorithm can be performed interchangeably on the cluster graph and the node graph.

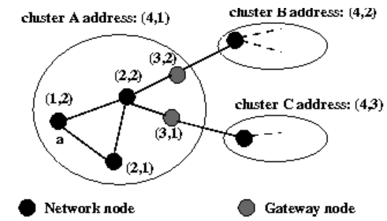
### C. Cluster-level state

Each cluster in our framework is not associated with a cluster head at the base level graph. Instead all members of a cluster maintain state that allows them to share the same knowledge about the cluster. This avoids the operational dependency on particular nodes as well as the need for additional election algorithms. Specifically, each node maintains the coordinates of its cluster in the embedding of the cluster graph, and the cluster adjacency information of neighbour clusters. The latter consists of the neighbour cluster coordinates alongside a respective inter-cluster gateway's coordinates. An inter-cluster gateway is any network node in a cluster that has at least one neighbour in another cluster.

Figure 1 exemplifies the overall neighbourhood information stored at each node for the base level and the additional cluster level. Note that the cluster-level information should not impose any scalability problems as it would sum up to $O(k)$ ($k$ being the number of neighbour clusters – independent of the network size, and related to the density of the graph). Moreover, one may integrate the additional state in the same adjacency tables lending to more memory savings in small cluster sizes, since a node neighbour at the base level is often a cluster gateway. An evaluation of the memory cost entailed (without this optimisation) is provided in Section IV-D.

### D. Operation of greedy routing

In order to enable greedy routing at the cluster level, the packet header has now been extended so as to contain both the destination node's coordinates in the base level embedding, as well as those of its cluster in the embedding of the cluster graph.




(a) Network and cluster addressing

Flat Level Neighbour Table

| |
|---|
| (2,2) |
| (2,1) |

Cluster Level Neighbour Table

| Cluster | Gateway |
|---|---|
| (4,2) | (3,2) |
| (4,3) | (3,1) |

(b) Neighborhood information at node a

Fig. 1. Network and cluster Graph embedding example

*1) Routing at the cluster level:* Given a destination cluster coordinates a forwarding node first computes the distance from its own cluster. It then selects among its neighbour clusters the one with the smallest distance to the destination's cluster. If the current cluster is closer to destination than the selected neighbour cluster, a stop condition is reached at the cluster level (in accordance to the basic greedy algorithm). In order to reach a selected neighbour cluster a node needs first to route greedily to the respective gateway node in its own cluster. Hence the gateway's coordinates are prepended in the packet that is greedily forwarded to it. Finally, at the neighbour cluster's gateway node the temporary coordinates are removed and forwarding towards the destination coordinates is resumed.

Note that a stop condition might also be reached in case the neighbour cluster's gateway is not greedily reachable. Such a phenomenon may occur if the clustering approach introduces concave void regions in the intra-cluster graph.

*2) Cross-level routing:* The benefits of clustering come into effect when one combines greedy routing at the cluster level with routing at the base level. We are currently considering two possibilities, in the first (top-down) greedy routing starts at the cluster-level and concludes at the base level, while in the other (bottom-up) routing starts at the base level and if a local minimum is encountered the algorithm tries to progress further at the cluster level.

In the top-down approach, even if the destination cluster is reached, routing has to always switch to the base level for intra-cluster delivery to the destination node. By contrast in the bottom-up approach, it is likely that routing is concluded at the base level (if no local minima are encountered) without switching to the cluster level graph.

A third possibility would be to continually alternate between the two levels, whenever a local minimum is encountered. However, unless some additional sophistication is introduced (constraints), this behaviour can lead to routing loops as

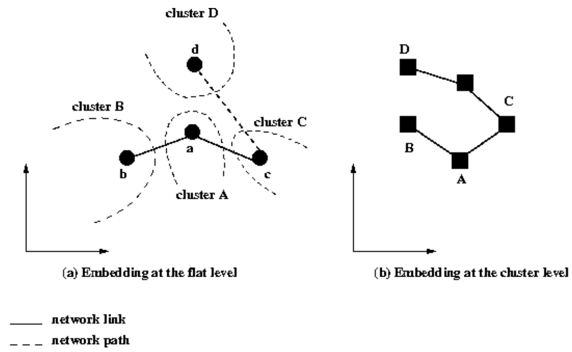



Fig. 2. Greedy routing loops between network levels

exemplified in Figure 2. Node $a$ which is a local minimum at the base level (Figure 2.a), wishes to greedily route to node $d$. By switching to the cluster level and using the cluster level embedding of Figure 2.b, greedy progression is possible and node $a$ forwards the packet to cluster $B$, i.e. at gateway node $b$. However, cluster $B$ is a local minimum at the cluster graph, and routing at this level reaches a stop condition. Switching back to the base level, will result in $b$ forwarding the packet back to $a$, and an infinite loop is created. Clustering in this case will not solve the problem. To avoid worsening the problem either, for now we forbid the switching between the two levels more than twice[1], by nullifying the destination cluster's address in the packet.

### E. Multiple cluster levels

The practise that we have described so far can be repeated to introduce additional cluster levels, by recursively clustering over the cluster graph. The greedy routing would function in a similar way to the two-level case, however, the gateway nodes between clusters at the third level are in fact clusters as well (of level 2).

The rationale is that while the first cluster level performs some degree of "averaging" over the network graph connectivity and "smoothing" the concave voids, the additional clustering further refines the abstract topology by averaging and smoothing among cluster groups. This would effect a more gradual (as opposed to sudden) bending of the (Euclidean) path between two nodes, however at the cost of higher *path stretch*. In our evaluation we experimented with up to two levels of clustering, in order to discover the degree of additional improvement compared to the cost of this process.

Another approach when experimenting with multiple cluster levels would be to use more than one cluster graphs in parallel, at the same level (each produced with different clustering method or the same method with a different seed). Using a copy and branch operation during the switch to the cluster level, greedy routing would then be carried out in both cluster graphs. The idea would be that while routing along one cluster graph might reach a stop condition, it may progress along the other. In case the packet gets delivered along both paths the destination would eliminate the copies. This approach however requires further investigation in regard to choosing the right action policy when stop conditions are met. We have not experimented yet towards this direction.

## IV. Evaluation

In this section we present the results of exploration with clustering. We report on the performance improvement observed (in finding greedy paths) compared to three reference geometric embeddings, namely NoGeo, classical MDS and metric MDS. We also examine the overhead of clustering regarding the additional per-node state and the average per-path routing stretch.

### A. Experimental set-up

All simulated topologies have been produced using the Unit Disk Graph approach [34], by randomly scattering nodes on a fixed size grid and establishing a link between two nodes whenever a minimum threshold distance exists between them. The number of nodes in the topologies range from 500 to 5000, incremented in 500 nodes steps. The tests are therefore repeated as a function of network size, where we keep the graph density constant: average node degree 7 (this degree exhibits a reasonable number of route failures in the base embeddings to allow comparison). We also conducted tests as a function of the network density, where we keep the network size fixed at 2000 nodes and vary the average node degree (by changing the Unit Disk radius).

The two MDS systems were based on the Matlab implementations, while the NoGeo system was based on a simulator running the network of springs method described in [20].

### B. Clustering algorithm used for the tests

As we are interested in exploring the effects of clustering in general rather than focusing on a specific clustering approach, we have used two diametrically opposite clustering methods in terms of sophistication. This should provide us with some boundary insights on the performance variability in regard to clustering method sophistication.

The first is a centralised spectral-clustering algorithm [32], [33] that tries to find optimal clusters following a relaxation of the normalized minimal cut problem. This method relies on knowledge of the graph's adjacency matrix[2]. The second is a simple distributed agglomerative clustering method that creates clusters in a stochastic manner. First, a "root" node initiates the cluster formation by broadcasting a *cluster-join* to its $k$-hop neighbours (where $k$ is set in advance). Nodes that are not already members of a cluster, upon receiving the invitation will join the cluster. Then a new root node is chosen in the proximity of the last formed cluster, and the operation repeats until all nodes are assigned to a cluster.

In all tests the generated cluster graph is embedded in the 3-dimensional Euclidean space.

[1] On-going work focused on addressing this challenge will be reported in the future, however it is beyond the exercise of this paper

[2] In order to meet our primary objective, in this case we had to relax our secondary requirement for distributed computation

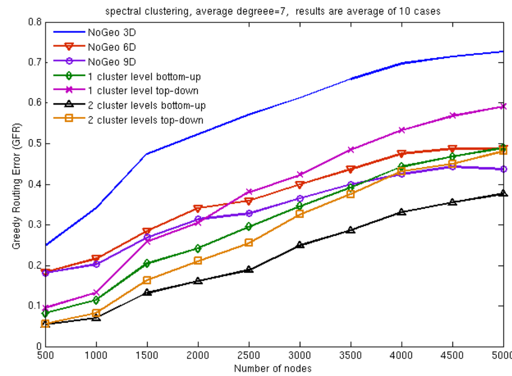



Fig. 3. NoGeo greedy route failures with spectral clustering

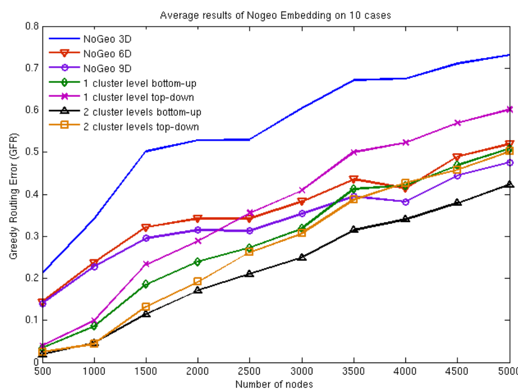



Fig. 4. NoGeo greedy route failures with agglomerative clustering

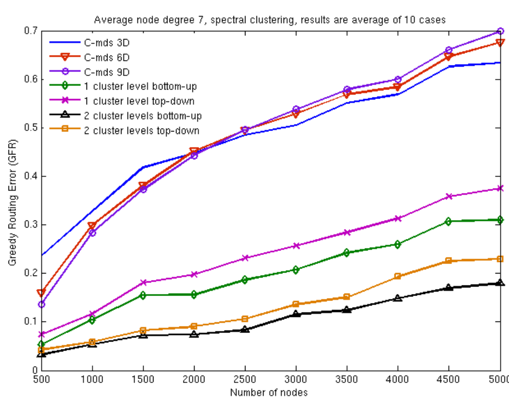



Fig. 5. Classical MDS embedding using spectral clustering

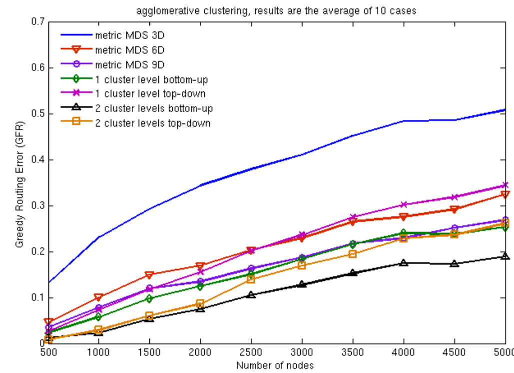



Fig. 6. Metric MDS Greedy routes failure using agglomerative clustering

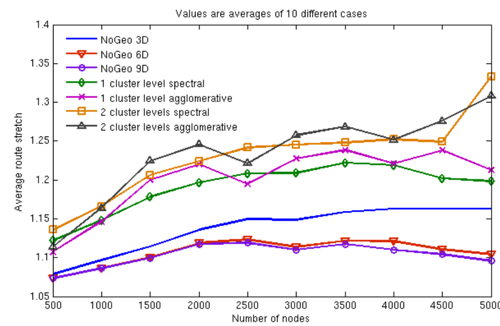



Fig. 7. Average stretch using NoGeo embedding

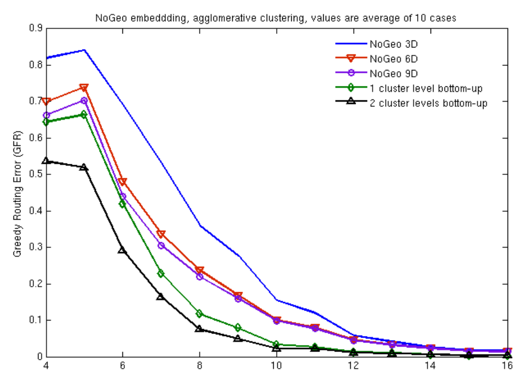



Fig. 8. Average node degree variations on a 2000 nodes network

### C. Greedy routing performance

To evaluate the routing performance, we forward greedily a packet from every node in the network to all other nodes ($N \times (N-1)$ routes for a network of $N$ nodes), and we monitor the percentage of those packets that failed to reach their destination. We call this metric *Greedy Failure Rate (GFR)*, and we plot it as a function of network size and network density. The results shown in the figures are averages over 10 runs with topologies of same characteristics (network size and density).

One first general observation regarding the cross-level routing, confirmed in all plots, is that the bottom-up strategy always yields better results than the top-down approach. This validates our intuition explained in section III-D2.

Starting with the NoGeo system, Figure 3 reports the GFR as a function of the network size, when the NoGeo embedding was used in combination with spectral clustering. Figure 4

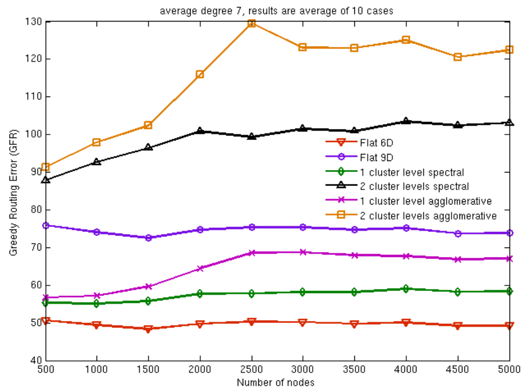



Fig. 9. Average storage overhead per node for 1 and 2 cluster levels

reports on the same test when the agglomerative clustering was used instead. In each of them, we practically evaluated two orthogonal aspects: one being the absolute performance improvement (GFR reduction) when we employ clustering over the base embedding, and the second is how this improvement compares to the performance of a more accurate base embedding in a higher dimensional space (3D, 6D and 9D curves in the figures).

One sees a notable improvement from the introduction of clustering, which for the various network sizes, ranges between 15% and 25% when one cluster level was used, and between 20% and 35% when two cluster levels were used. An interesting observation is that between agglomerative and spectral clustering there is no substantial difference in the average GFR, which suggests that the measured effects might be statistically quantifiable for different embedding systems, irrespective of clustering method. However, such a hypothesis needs more thorough examination with additional clustering methods. The variance in the datasets (may be perceived in the smoother curves) for the GFR, was smaller in the case of spectral clustering, most likely because of its less stochastic nature (in generating cluster graphs).

Looking at the groups of curves in the same two Figures, one can observe an asymptotic reduction of the GFR as the dimensionality of the base embedding (NoGeo) increases from 3D to 9D, and which is further "magnified" in larger networks. An analogous pattern seems to exist as the number of cluster levels increases in all configurations. A noticeable difference, between these two groups, is that the NoGeo lines are curved with a monotonically decreasing angle, compared to the more linear clustering performance curves. As a result although clustering yields better results than high dimensional embeddings at the beginning, as the network size increases (with a fixed density), a high-dimensional embedding scales better.

Analogous improvements and observations are reported in Figures 5 and 6, in comparison to the classical MDS and metric MDS base embeddings respectively. As the results were similar in all configurations, to save space we only show metric MDS with agglomerative clustering, and classical MDS with spectral clustering. An additional observation here, in the case of classical MDS (Figure 5), is that clustering performs substantially better. On the other hand, increasing the dimensionality of the base embedding does not noticeably improve the performance, rather after a certain network size it actually worsens it!

Looking at the consequent average route stretch factor (Figure 7) of the successful greedy routes, as anticipated (and conformed also in [10]) the different clustering methods, transparently from the greedy algorithm, curve the Euclidean paths in order to get round the local minima. Yet, the increase of the stretch factor is quite small, i.e. $\leq 0.15$ of the respective stretch of the 6D and 9D NoGeo, and $\leq 0.09$ of the 3D NoGeo.

Regarding the network density, Figure 8 reports on the performance of the same configurations as before, as a function of the network density in the case of a NoGeo embedding. Clustering in this case provides a significant performance improvement in networks with an average degree below *11*. Beyond this threshold, clustering still performs better, however with a minimal improvement, as the performance of all configurations converges asymptotically to near-zero GFR.

### *D. State overhead for clustering*

To assess the overheads introduced by using our framework for supporting clustering in geometric routing, we assume the following simple reference model for comparisons: For addressing across $l$ hierarchically organised cluster levels, each embedded in a $d$ dimensional space, a node's addressing information occupies $l \times d$ units in the packet header. This is required to enable greedy routing decisions and distance calculations at all different levels (1 unit comprising the space requirement for a coordinate, in one dimension).

To increase on the other hand the performance of greedy routing using an embedding approach (without clustering), one can increase the dimensionality of the embedding space. An increasing of the dimensionality to $d' = (l \times d)$, would impose the same addressing space requirements as $l$ clusters (same packet overhead, and each neighbour record in the adjacency matrix of a node, would occupy the same amount of space). This analogy will be the basis for our memory overheads comparisons (multiple clustering levels versus a high dimensional base embedding). It practically means that we would like to compare 1-level clustering regime with a 6D base embedding of the network graph, and 2-level clustering with a 9D base embedding of the network graph.

We then calculate the space overheads of the different clustering configurations evaluated earlier as follows: For a $d$-dimensional base embedding of a network graph with approximate density (node degree) $k$, the average amount of adjacency state maintained in each node is $(k + 1) \times d$. In the case of clustering with our framework, using $l$ cluster levels, the same $d$-dimensional embedding across all level, and an average node degree $k_i$ for the graph in level $l_i$, the amount of adjacency state maintained in each node totals $l \times d + \sum_{i=1}^{l}(2 \times k_i \times d)$ (multiplier 2 represents the fact that for each cluster adjacency record a node needs to store both the cluster address and a respective gateway address).

Figure 9 reports the resulting averaged estimates over the 10 topologies, for each of the configurations discussed in the previous section. In all cases but one, the memory cost does not vary significantly as the network sizes increases. The exception were the configurations where agglomerative clustering was used. The exception of agglomerative clustering can be explained by the variability in the clustering performance for the agglomerative methods, which leads to cluster graphs with different average node degrees. This result suggests that the choice of clustering method has a variable impact, but a careful choice of clustering method can retain the scalability in regard to network sizes.

A comparative examination of the plots reveals that the overhead of one level of spectral clustering when compared to a 6D embedding would amount to having each node store less than two additional 6D neighbour addresses (10 additional unit coordinates). In the case of a 2-leveled spectral clustering compared to a 9D embedding, the overhead amounts to storing three additional 9D addresses. On average, one sees an overhead of approximately 8-15% in the case of spectral clustering and 15-35% in the case of agglomerative clustering. This overhead becomes 15-35% and 20-70% respectively, when comparing 2-level clustering with the 9-dimensional base embedding. At the same time the cost from a 6-dimensional base embedding to a 9-dimensional one increased by 50%. Therefore, roughly we may say that the overhead of adding clustering levels increases twice as fast as the cost of the corresponding dimensionality increase in the base embedding.

## V. Discussion

### A. On the general effects of clustering

Clustering in greedy geometric routing has the potential of indirectly creating two interesting effects, which motivated our work in this paper. Both of these effects are confirmed by our results.

The first is on void regions that are the product of an embedding of the network graph in a low dimensional space. On one hand the cluster graph essentially "thins" the densely connected regions of the network graph, which results in a more uniformly connected and spread-out sub-graph. Thinning happens as the clustering process "packs" these regions inside cluster nodes. As a result of this the routing process is factored in two levels: (a) within clusters where the dense connectivity is more likely to yield successful routing under a greedy regime, and (b) across clusters whose more uniform connectivity and arrangement in their coordinate space reduces areas of local minima and makes greedy routing more effective. From a macroscopic perspective the overall effect can be seen as a thinning and zooming out process that dilutes void regions.

On the other hand clusters function as attractors that curve the trajectory of routing paths. To understand how this works imagine that in greedy routing the destination applies a pulling force on the packet, which in every forwarding decision pulls the packet a little bit closer. Then every time routing decisions are transposed to the cluster level a second force component (at the direction of the cluster) is temporarily added to the destination pulling force, influencing the resulting motion of the packet along the norm of the net addition of the two components. As this happens implicitly and still under a greedy regime, it is significantly less complex and problematic than face or other hybrid routing methods for recovering from local minima[3]. This latter effect, which is a direct consequence of the first, has also been pointed in [10].

The confirmation of the positive impact that these effects have on greedy routing has inspired some ideas for follow up work. The first relates to resolving the problem of routing loops in a regime that would allow continuous hopping between embedding levels (see section III-D2). This practically would make the benefits from the first effect more pronounced. One possibility towards this direction may be to consider an in-packet path history mechanism similar to the one used in [19].

A second interesting exploration is to embed the nodes of the cluster graph in the same coordinate space as the base embedding as virtual nodes. Given the effect of clusters to function as path attractors, the explored hypothesis is to see if the assumed locations of virtual nodes can influence greedy routing decisions away from links that lead to local minima.

One additional exploration would be to compare the performances of our approach with combinations of different embeddings of the network graph that are not necessarily cluster views of it. One could imagine routing on two different embeddings of the base level graph, referring to one if a dead-end is reached on the other. This would be somehow similar to the notion of different *realities* in [35]. In the same way one could picture a comparison with a three-level routing in which the second and third levels are both embeddings of a cluster view of the network graph using different clustering methods.

### B. Clustering versus inter-domain routing

Drawing on the similarity of the two level identifiers (cluster level, node level) with network and host identifiers in the Internet, one may be tempted to think that there is a similarity between the work we present here and hierarchical routing techniques [36] or inter-domain routing [37] in the Internet. However, this similarity is neither conceptual, nor functional. Network hierarchies in classical and inter-domain routing serve as a measure to aggregate and reduce the amount of control information required for the operation of stateful routing. No such need exists in geometric routing. Functionally, another major difference is that our proposal considers routing in both directions of the hierarchy (from cluster level to node level, as well as from node level to cluster level). In classical and inter-domain routing approaches, routing always starts at the aggregate level, and concludes at the host level.

[3]Although to conclude the absolute superiority we would have to design a suitable routing performance benchmark for this comparison

## VI. Conclusion

In this paper we have provided some insights on the effects of clustering for improving the performance of greedy routing. We looked at clustering from a general perspective (as opposed to focusing on a specific method), and we proposed a framework that enables cluster based greedy-routing over any low-dimensional embedding. In our evaluation, with clustering methods of varying sophistication, we measured and reported significant improvement of up to 25% (1-level clustering) and 35% (2-level clustering), over a number of reference systems in the literature. We experimented with a variety of configurations (different network sizes and densities), so as to develop a better understanding on the effects of clustering, and quantify its overheads over the stateless nature of geometric routing. Finally, based on the insights we acquired, we have identified a number of interesting directions for further exploration and follow work.

## References


[1] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, Boston, USA, 2000.

[2] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with Guaranteed Delivery in Ad-hoc Wireless Networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, November 2001.

[3] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: of theory and practice," in *Proceedings of the 22nd annual ACM symposium on Principles of Distributed Computing, PODC '03*, Boston, Massachusetts, USA, 2003.

[4] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th international ACM symposium on Mobile Ad-hoc Networking & Computing, MobiHoc '03*, Annapolis, Maryland, USA, 2003.

[5] L. Doherty, K. pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '01*, Anchorage, Alaska, USA, April 2001.

[6] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000.

[7] L. Tang and M. Crovella, "Virtual landmarks for the internet," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, IMC '03*, Miami Beach, FL, USA, 2003.

[8] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," in *Proceedings of the 12th annual international ACM conference on Mobile Computing and Networking, Mobicom '06*, Los Angeles, CA, USA, 2006.

[9] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," in *Proceedings of the 6th international workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, DIALM '02*, Atlanta, Georgia, USA, 2002.

[10] Q. Fang, J. Gao, L. Guibas, V. de Silva, and L. Zhang, "GLIDER: gradient landmark-based distributed routing for sensor networks," in *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '03*, March 2005.

[11] N. Carlsson and D. L. Eager, "Non-Euclidian geographic routing in wireless networks," *Ad-hoc Networing*, vol. 5, pp. 1173–1193, September 2007.

[12] N. Arad and Y. Shavitt, "Minimizing Recovery State in Geographic Ad-hoc Routing," *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 203 –217, February 2009.

[13] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation, NSDI'05 - Volume 2*, Berkeley, CA, USA, 2005.

[14] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad-hoc networks in city environments," in *Proceedings of IEEE Symposium on Intelligent Vehicles*, June 2003.

[15] K. Seada, A. Helmy, and R. Govindan, "Modeling and analyzing the correctness of geographic face routing under realistic conditions," *Ad-hoc Networking*, vol. 5, pp. 855–871, August 2007.

[16] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York, USA, 2005.

[17] C. H. Papadimitriou and D. Ratajczak, "On a conjecture related to geometric routing," *Theoretical Computer Science*, vol. 344, no. 1, pp. 3–14, 2005.

[18] G. R. Hjaltason and H. Samet, "Properties of Embedding Methods for Similarity Searching in Metric Spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 530–549, 2003.

[19] R. Fonseca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, and I. Stoica, "Beacon vector routing: scalable point-to-point routing in wireless sensornets," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, NSDI'05*, Boston, USA, 2005.

[20] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proceedings of the 9th annual international conference on Mobile computing and networking, MobiCom '03*, San Diego, USA, 2003.

[21] S. Oh, A. Montanari, and A. Karbasi, "Sensor network localization from local connectivity: Performance analysis for the MDS-MAP algorithm," in *Information Theory Workshop*, Dublin, Ireland, January 2010.

[22] J. Kruskal and M. Wish, "Multidimensional scaling," Sage Publications, Beverly Hills and London, Tech. Rep., 1978.

[23] Y. Shavitt and T. Tankel, "Big-bang simulation for embedding network distances in euclidean space," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 993 – 1006, December 2004.

[24] R. Dhandapani, "Greedy drawings of triangulations," in *Proceedings of the 19th annual ACM-SIAM symposium on Discrete Algorithms*, San Francisco, USA, 2008.

[25] A. Moitra and T. Leighton, "Some results on greedy embeddings in metric spaces," in *Proceedings of the 49th Annual Symposium on Foundations of Computer Science, FOCS*, 2008.

[26] R. Kleinberg, "Geographic routing using hyperbolic space," in *Proceedings of the 26th IEEE International Conference on Computer Communications, INFOCOM' 07*, Anchorage, Alaska, USA, May 2007.

[27] D. Eppstein and M. Goodrich, "Succinct greedy graph drawing in the hyperbolic plane," in *Graph Drawing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5417, pp. 14–25.

[28] R. Sarkar, X. Yin, J. Gao, F. Luo, and X. D. Gu, "Greedy routing with guaranteed delivery using ricci flows," in *Proceedings of the International Conference on Information Processing in Sensor Networks, IPSN '09*, San Francisco, USA, April 2009.

[29] S. E. Schaeffe, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27 – 64, 2007.

[30] C. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265 –1275, September 1997.

[31] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *22nd Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '03*, San Francisco, USA, April 2003.

[32] C. Gkantsidis, M. Mihail, and E. Zegura, "Spectral analysis of internet topologies," in *22nd Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '03*, San Francisco, USA, April 2003.

[33] F. R. K. Chung, *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, Feb. 1997.

[34] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165 – 177, 1990.

[35] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '01*, San Diego, USA, 2001.

[36] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks," *Computer Networks*, vol. 1, pp. 155 – 174, 1977.

[37] M. Caesar and J. Rexford, "BGP routing policies in ISP networks," *IEEE Network*, vol. 19, no. 6, pp. 5 – 11, 2005.