Geographic Services for Wireless Networks

Karim Seada, Ahmed Helmy
Electrical Engineering Department, University of Southern California
{seada, helmy}@usc.edu

1. Introduction

In wireless ad hoc and sensor networks, building efficient and scalable protocols is a very challenging task due to the lack of infrastructure and the high dynamics. Geographic protocols, that take advantage of the location information of nodes, are very valuable in these environments. The state required to be maintained is minimum and their overhead is low, in addition to their fast response to dynamics. In this chapter, we present a state-of-the-art overview of geographic protocols providing basic functions such as geographic routing, geocasting, service location, and resource discovery. We introduce also some of our work on assessing and improving the robustness of geographic protocols to non-ideal realistic conditions corresponding to the real-world environments.

Geographic protocols are very promising for multihop wireless networks. These protocols take advantage of the location information of nodes to provide higher efficiency and scalability. In wireless environments, the locations of nodes correspond to their network connectivity, which makes geographic protocols natural components in these environments and it is expected that they will become major elements for the development of these networks. For obtaining the location information, different kinds of localization systems exist such as GPS, infrastructure-based localization systems, and ad-hoc localization systems.

Examples of multihop wireless networks are ad hoc networks and sensor networks. Ad hoc networks are infrastructure-less dynamic networks that could be an extension or alternative to infrastructure wireless networks, especially in situations where it is difficult or time-critical to deploy an infrastructure such as in disaster recovery or military applications. Commercially it could also be used to build small fast networks for conferences and meetings, vehicle networks, rooftop networks, or to extend the services provided by the cellular infrastructure. Sensor networks are networks of small embedded low-power devices that can operate unattended to monitor and measure different phenomena in the environment. Sensor networks are suited for applications such as habitat monitoring, infrastructure protection, security, and tracking.

We consider basic geographic protocols at the network layer: geographic routing, geocasting and geographic rendezvous mechanisms. Geographic routing provides a way to deliver a packet to a destination location, based only on local information and without the need for any extra infrastructure, which makes geographic routing the main basic component for geographic protocols. With the existence of location information, geographic routing provides the most efficient and natural way to route packets comparable to other routing protocols. Geocasting is the delivery of packets to nodes within a certain geographic area. It is an extension to geographic routing where in this case the destination is a geographic region instead of a specific node or point. Geocasting is an important communication primitive in wireless ad hoc and sensor networks, since in many applications the target is to reach nodes in a certain region. In geographic-based rendezvous mechanisms, geographical locations are used as a rendezvous place for providers and seekers of information. Geographic-based rendezvous mechanisms can be used as an efficient means for service location and resource discovery in ad hoc networks. They can also provide efficient data dissemination and access in sensor networks.

In the rest of this chapter, we will go through the basic geographic mechanisms: routing, geocasting, and geographic rendezvous. In Section 2 we discuss geographic routing protocols and some important related problems: the determination of destination location, the effect of location inaccuracy, and the effect of lossy links. In Section 3 we present the different geocasting mechanisms. In Section 4 we explain several geographic rendezvous mechanisms used for service location, resource discovery, and data access. Finally, the conclusions are presented in Section 5.

2. Geographic Routing

Routing in ad hoc and sensor networks is a challenging task due to the high dynamics and limited resources. There has been a large amount of non-geographic ad hoc routing protocols proposed in the literature that are either proactive (maintain routes continuously) [48], reactive (create routes on-demand) [31][47][49] or a hybrid [21]. For a survey and comparison see [52][10]. Non-geographic routing protocols suffer from a huge amount of overhead for route setup and maintenance due to the frequent topology changes and they typically depend on flooding for route discovery or link state updates, which limit their scalability and efficiency.

On the other hand, geographic routing protocols require only local information and thus are very efficient in wireless networks. First, nodes need to know only the location information of their direct neighbors in order to forward packets and hence the state stored is minimum. Second, such protocols conserve energy and bandwidth since discovery floods and state propagation are not required beyond a single hop. Third, in mobile networks with frequent topology changes, geographic routing has fast response and can find new routes quickly by using only local topology information.

In the discussion of routing mechanisms in Section 2.1, we have the following assumptions:

- Each node knows its geographic location using some localization mechanism. Location-awareness is essential for many wireless network applications, so it is expected that wireless nodes will be equipped with localization techniques. Several techniques exist for location sensing based on proximity or triangulation using radio signals, acoustic signals, or infrared. These techniques differ in their localization granularity, range, deployment complexity, and cost. In general, many localization systems have been proposed in the literature: GPS (Global Positioning System), infrastructure-based localization systems [63][50], and ad-hoc localization systems [11][53]. For an extensive survey of localization refer to Hightower *et al.* [27].

- Each node knows its direct neighbors locations. This information could be obtained by nodes periodically or on request broadcasting their locations to their neighbors.
- The source knows the destination location. In Section 2.2 we will discuss in more detail how this information could be obtained.

2.1 Routing Mechanisms

In geographic routing, each node knows the location of its direct neighbors (neighbors within its radio range). The source inserts the destination location inside the packet. During packet forwarding, each node uses the location information of its neighbors and the location of the destination to forward the packet to the next-hop. Forwarding could be to a single node or to multiple nodes. Forwarding to multiple nodes is more robust and leads to multiple paths to the destination, but it could waste a lot of resources (energy and bandwidth) and thus forwarding to a single node is more efficient and it is the common approach among unicast protocols. A main component in geographic routing is greedy forwarding, in which the packet should make a progress at each step along the path. Each node forwards the packet to a neighbor closer to the destination than itself until ultimately the packet reaches the destination. If nodes have consistent location information, greedy forwarding is guaranteed to be loop-free.

Takagi and Kleinrock [60] is an early work that presented the Most Forward within R (MFR) routing model, where R is the transmission radius. In MFR, a node transmits to the neighbor that

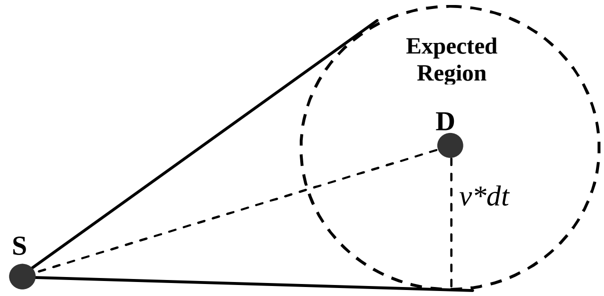

Figure 1: Source S sends the packet to all nodes in the direction of destination D expected region, where *v* is the velocity of D and *dt* is the time since last location update

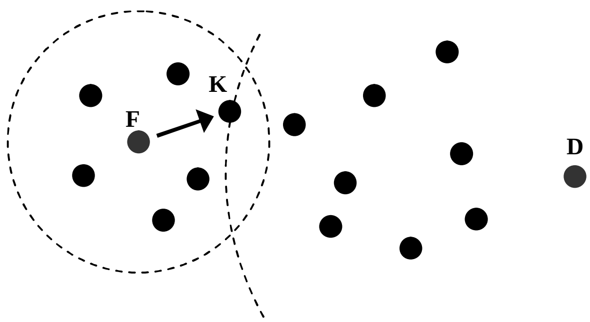

Figure 2: Greedy forwarding: Node F forwards the packet to neighbor K, which is the neighbor closest to the destination D

provides the maximum progress in the direction of the final destination, in order to minimize the number of hops between the source and the destination. The objective of that work was to obtain the optimum transmission radius in a content-based channel. In 1987, Finn [16] proposed Cartesian routing as a scalable routing solution to interconnect isolated LANs in the Internet. Each node forwards the packet to the neighbor closest to the destination among its neighbors that are closer to the destination. In [28] and [44], Imielinski and Navas proposed integrating geographic coordinates into IP to enable the creation of location dependent services in the Internet. They presented a hierarchy of geographically-aware routers that can route packets geographically and use IP tunnels to route through areas not supporting geographic routing. Geographically-aware routers can determine which geographic areas they are servicing and based on that information and the packet destination area, each router, when it receives a packet, decides whether it services that destination area or it should forward the packet to its parent or to some of its children in the hierarchy.

In [1], Akyildiz *et al.* used the reported geographic location of a mobile host to perform selective paging in cellular networks by paging a set of cells around that location. Among the earliest work to consider the geography for routing in ad hoc networks is LAR [35] by Ko and Vaidya, which uses the location information of nodes for route discovery and not for data delivery. LAR improves the performance of non-geographic ad hoc routing protocols by limiting discovery floods to a geographic area around the destination expected location. DREAM [5] is a routing protocol that uses the location information for data delivery in ad hoc networks. The packet is forwarded to all nodes in the direction of the destination. Based on the destination location and its velocity, the source determines an expected zone for the destination and forwards the packet to all nodes within an angle containing the expected zone. If the sender has no neighbors in the direction of the destination, a recovery procedure using partial flooding or flooding is invoked. Figure 1 shows an example for directional flooding which could be used for route discovery in LAR or data delivery in DREAM. In Compass routing [38], a node forwards the packet to the neighbor whose edge has the closest slope to the line between that node and the destination; that is the neighbor with the closest direction to the destination. Compass routing is not guaranteed to find a path if one exists.

Bose *et al.* [9] and GPSR [33] present the common form of greedy forwarding in ad hoc networks. Packets contain the position of the destination and nodes need only local information about their position and their immediate neighbors' positions to forward the packets. Each wireless node forwards the packet to the neighbor closest to the destination among its neighbors (within radio range) that are closer to the destination as shown in Figure 2.

Greedy forwarding is very efficient in dense uniform networks, where it is possible to make progress at each step. Greedy forwarding, however, fails in the presence of voids or dead-ends, when reaching a local maximum, a node that has no neighbor closer to the destination (Figure 3).

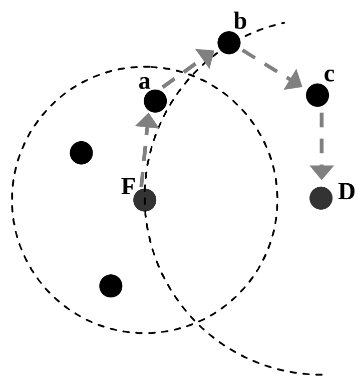

Figure 3: Greedy forwarding fails at node F when there are no neighbors closer to the destination D, although a path through a farther neighbor F-a-b-c-D exists

In this case, it will fail to find a path to the destination, even though paths to the destination through farther nodes may exist. Previous protocols deals with dead-ends in different ways. In MFR [60], if no progress could be made in the forward direction, the dead-end node sends the packet to the least backward neighbor, which is the neighbor closest to the destination among its neighbors. This could cause looping and nodes need to detect when they get the same packet for a second time. Finn [16] proposed using limited flooding for a number of hops to overcome dead-ends. When a node is reached that has no neighbors closer to the destination, it sends a search packet for *n* hops away. Closer nodes to the destination reply back and the closest node to the destination among those nodes is chosen to forward the packet. The value of n is set based on the topology structure (estimated size of voids) and the desired degree of robustness. LAR and DREAM, which use directional flooding, did not provide specific mechanisms to deal with voids

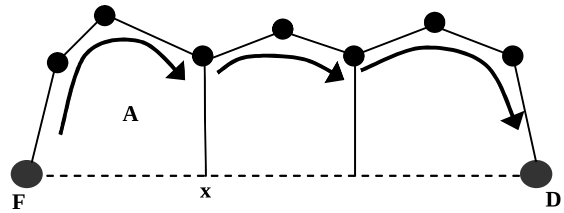

Figure 4: Face (Perimeter) routing: The packets traverse planar faces between a node F and the destination D using the right-hand rule

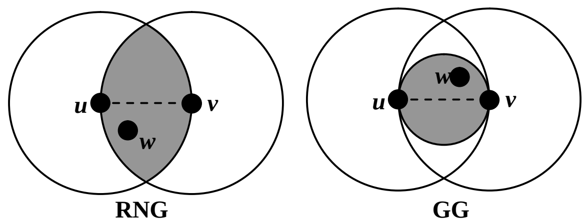

Figure 5: Local and distributed planarization algorithms. Node *u* removes the edge *u*-*v* from the planar graph, if a witness *w* exists

that stop the flood before reaching the expected zone. It is assumed that global flooding will be used as a recovery if directional flooding fails. De Coute [14] shows a probabilistic approach that uses intermediate node forwarding to overcome dead-ends. When greedy forwarding fails, the source picks a random intermediate point and routes the packet though it to the destination. The random point is picked randomly from an area between the source and the destination. The area is increased each time the routing fails.

The previous approaches for dead-end recovery do not guarantee that the packet reaches the destination if a path exists (unless global flooding is used, which causes large overhead). A local algorithm called Compass Routing II was presented in [38], which guarantees that the packet reaches the destination. Compass Routing II, which becomes known as face routing or perimeter routing, works in planar[1] unit[2] graphs by traversing the faces intersecting the line between the

[1] Planar graphs are graphs containing no cross links.
[2] In a unit graph a pair of nodes is connected if and only if the distance between them is below a certain threshold which is the radio range in this case.

source and the destination consecutively until reaching the destination as shown in Figure 4. Bose et al. [9] presented algorithms and proofs for extracting planar graphs from unit graphs and for face routing in the planar graphs to guarantee delivery. Due to the inefficient paths resulting from face routing, they proposed combining face routing with greedy forwarding to improve the path length. Face routing is used when greedy forwarding fails until a node closer to the destination is reached, then greedy forwarding could be resumed again. This way the algorithm will remain loop-free. In order for face routing to work correctly, a planar connectivity graph for the network needs to be constructed and so a planarization algorithm is required to create the planar graph. In Figure 5, RNG [62] and GG [17] are examples of algorithms that create a planar graph from the non-planar physical topology by selecting a subset of the links and using only those links during face routing. A desirable feature in these algorithms is that they are local (a node needs to know only its own and neighbors' locations) and run in a distributed manner, so that each node can decide the links to include for planar routing using only local information independent of other nodes. The main idea of both algorithms is for a node to exclude an edge to a neighbor from the planar graph if there is another path through a different neighbor called *witness*. The witness should exist in a specific intersection area between the two nodes of the edge. In [4], a variant of face routing is described that is more robust to irregular transmission ranges and can tolerate up to 40% of variation in the transmission range at the cost of a limited amount of extra overhead.

In summary, greedy forwarding alone *does not* guarantee the delivery of packets because of dead-ends (variously called local maxima or voids). Face routing on a planar graph theoretically does guarantee the delivery of packets. For improved performance, face routing is integrated with greedy forwarding and is used as a way to overcome dead-ends when greedy forwarding fails. Wireless network connectivity is in general non-planar, this is why the planarization component is required to create a planar graph by using only a subset of the physical links during face routing. Face routing, similar to greedy forwarding, is also stateless and nodes need to keep only information about their direct neighbors in order to forward a packet, thus combined geographic protocols of greedy and face routing are stateless. Greedy forwarding coupled with face routing is the common efficient approach of the currently proposed geographic protocols.

GPSR [33] is a geographic routing protocol for wireless networks that combines greedy forwarding and face routing (perimeter routing). Packets contain the position of the destination and nodes need only local information about their position and their immediate neighbors' positions to forward the packets. Each node forwards the packet to the neighbor closest to the destination using greedy forwarding. When greedy forwarding fails, face routing is used to route around dead-ends until closer nodes to the destination are found. In Figure 4, node *F* is forwarding a packet using face routing to node *D*. Using the right-hand-rule the packet starts traversing face *A*, switching to other faces intersecting *FD* until reaching the face containing *D*. In [33], packet-level simulations using 802.11 wireless MAC layer and comparisons with an ad hoc routing protocol, DSR, are provided. GOAFR [39] is another protocol proposed later that also combines greedy forwarding with face routing and is designed to be both asymptotically optimal and average-case efficient. GOAFR achieves worst-case optimality (analytically proved) of the path length by using limited elliptic regions for face routing and recursively increasing the ellipse size until finding a close-to-optimal path. This could improve the efficiency in low-density networks.

Other approaches for geographic routing have also been presented. In Gao *et al.* [18], a clustering algorithm is used to group nodes into clusters and then a planar graph called a

Restricted Delaunay Graph (RDG) is built between the cluster-heads. RDG can be used as an underlying graph for geographic routing and it has the benefit that the path length between two nodes is a constant factor from the optimum length. Gao [18] shows that routing on RDG graphs outperforms graphs built by RNG or GG, but maintenance for the clusters and graph is required. Terminode Routing [7] presents a different approach by dividing routing into two levels and using geographic routing for remote routing and a distance vector protocol for local routing. Geographic routing is used for routing to remote destinations to provide scalability in large mobile ad hoc networks, but as the packet arrives close to the destination (2 hops away) local routing is used to avoid inconsistencies in the destination location. In Terminodes, a protocol called Anchored Geodesic Packet Forwarding is used for geographic routing, where the source defines a set of anchors (fixed geographic points) in the path to the destination. The goal of using anchored paths is to try to avoid obstacles and gaps by setting anchors accordingly. A packet is sent through the anchors to the destination; each node forwards the packet towards the next anchor in the list using a greedy approach, until the packet arrives to a node in proximity of this anchor, then the next anchor in the list is used and so on. A path discovery method is proposed to learn about anchors in the path. If no anchors are known, the destination location is used as the next anchor. Jain *et. al.* [30] uses another approach which is a mix of greedy forwarding and traditional ad hoc routing. Each node maintains a routing table containing its direct neighbors and their positions, which it uses for greedy forwarding. When a packet reaches a dead-end, a route discovery protocol is initiated to find a path to the destination. Each node along the path set an entry in its routing table for the next-hop to that destination. For route discovery no explicit algorithm is specified; flooding, depth first search, face routing, or distance vector routing could be used to learn the path.

In sensor networks communication is typically data-centric, which means communication between nodes is based on the content of data rather than the specific node identities. Messages are directed for named data instead of named nodes. Directed diffusion [29] is a data-centric communication approach presented for sensor networks. In directed diffusion data are named by attribute-value pairs and nodes interested in the data diffuse their interest to other nodes. Data can then be forwarded along the reverse path to the interested nodes. In [25], different diffusion algorithms were discussed (e.g. push or pull) to design the protocol based on the application characteristics. Using geographic information to limit the diffusion by geographically scooping the messages was also presented. GEAR (Geographical Energy Aware Routing) [68] is an energy aware geographic protocol designed with the goal to increase the lifetime of sensor networks. GEAR uses energy aware metrics for neighbor selection in such a way that each node tries to balance the energy consumption among its neighbors using only local information by maintaining a cost function for each neighbor computed based on its location and an estimation for the energy consumed by that neighbor. GAF (Geographical Adaptive Fidelity) [65] uses the geographic information for energy conservation by building a geographical grid, such that only a single node needs to be turned on in each cell and other nodes are turned off. The cell size is set based on the radio range of nodes so that all nodes in the cell are equivalent from a routing perspective. TTDD (Two-Tier Data Dissemination) [67] provides a different way for data dissemination than directed diffusion. Instead of the sink propagating queries to all nodes and sources replying back; in TTDD each source builds a grid structure and sets its forwarding information at the nodes closest to the grid points, so that queries from the sink traverse only nodes in the local cell and some grid nodes towards the source. TTDD uses geographic greedy forwarding to construct and maintain the grid. This approach is beneficial when sinks are mobile, since location updates are propagated within the local cell and some grid nodes instead of the whole network.

SPEED [24] is a geographic routing protocol designed for real-time communication in sensor networks. SPEED handles congestion and provides soft real-time communication by using feedback control and non-deterministic geographic forwarding. It also provides a different way to handle dead-ends similar to the way it handles congestion. Non-deterministic geographic forwarding is used to balance the load among multiple routes. A node computes a relay speed to each of its neighbors by dividing the advance in distance to the destination by the estimated delay to forward the packet to that neighbor. The node then forwards the packet to a neighbor closer to the destination that has a speed higher than a certain threshold with a probability based on that neighbor speed compared to other neighbors. If no neighbor has a speed higher than the desired speed, a neighborhood feedback loop determines whether to drop the packet or reroute it in order to reduce the congestion. Backpressure rerouting is used to avoid both congestion and dead-ends by sending a backpressure beacon to the upstream node. The upstream node will try to forward the packet on a different route or further backpressure will occur until a route is found. Dead-end recovery using this backpressure mechanism does not guarantee to find a path. SPEED considers also others functions such as geocast and anycast which can be activated after the packet enters the destination area.

A related approach to geographic routing is trajectory-based forwarding [46], a method presented to route packets along curves in dense sensor networks. In this method, the trajectory is set by the source and intermediate nodes forward the packet to nodes close to the trajectory path.

2.2 Destination Location

In the previous section we have mainly focused on the routing problem and assumed that the packet destination location is known to the source. How the destination location is obtained is a separate problem that in many cases depends on the application. Most of the routing protocols discussed have not considered this problem explicitly. In many applications in ad hoc and sensor networks, the node ID itself is irrelevant and nodes are characterized by their location. In those applications packets do not need to be forwarded to specific nodes and a node close to the destination location or in a certain area around the destination can process the packet. For example, in sensor networks, queries may be sent to specific locations that the access point decides based on previous events and measurements. In geocasting, packets are sent toward regions and all nodes in the region can receive the packet.

In applications where the packet should be sent to a specific node, a mapping between the node ID and its current location is required. The source needs to obtain the destination current location before forwarding the packet, e.g. by consulting a node location service. It is important for the location service to be efficient and at the same time consistent with node locations. A simple way to obtain node locations is by having nodes propagating their locations through the network and other nodes storing these locations. This approach causes large energy and bandwidth overhead, especially with node mobility, and the storage will be high since each node stores the locations of all other nodes, even if it may not need most of them. Another approach is to flood queries that search for the destination location and the destination can reply back with its current location. Approaches based on global flooding do not scale to large networks. DREAM [5] considered the problem of locating destinations and provided a solution based on location propagation. In order to limit the overhead, nodes propagate their locations based on two observations: the distance effect, where updates are propagated as a function of the distance between the node updating its location and the node receiving the update in such a way that closer nodes receive more updates and have more accurate information about a destination location. The second observation is that each node sets the frequency of location updates based on its mobility rate, so that low mobility nodes send fewer updates.

A different approach that avoids flooding is to use location servers that keep track of node locations. Nodes moving send only to these servers to update their locations and other nodes can query the servers to obtain the recent locations. In infrastructure-based networks (e.g. in Mobile IP and in cellular networks) centralized fixed well-known servers provide this service, but in ad hoc and sensor networks it is difficult to use centralized servers due to the lack of infrastructure and due to the topology changes and dynamics. In Terminodes [6], each node has a Virtual Home Region (VHR) that is known or can be computed by other nodes. Each Node updates its location by sending the location update to its VHR. All nodes in the VHR will store that node location. Queries for a node location will be sent to the node VHR, where the nodes there can reply back. In [64] also, each node has a home region and all nodes in the home region store its location. GLS [41] presented a scalable distributed node location service for ad hoc networks. Each node updates a small set of location servers with its location. The node uses a predefined geographic hierarchy and a predefined ordering of node identifiers to determine its location servers. At each level of the hierarchy, the node chooses a location server as the node, from the corresponding region of that level, that has the closest ID to itself. Queries use the same hierarchy and identifier ordering to access a location server. In [66], a geographic hierarchy is also used to map each node to location servers at different levels in the hierarchy such that the location is represented with different accuracy at each level. Instead of choosing a location server based on its node ID as in GLS, a mapping function is used to map the destination ID to one of the minimum partitions in the hierarchy and choose a node that covers this partition as the location server. The location of a node is stored by its location servers at different levels of accuracy, such that further location servers store approximate locations while closer servers have more accurate locations. This way a smaller number of location servers will need to be updated when the node moves which reduces the overhead due to node mobility. Queries for the destination location will start with approximate regions and obtain more accurate information about the location as they get closer to the destination. A scheme that uses uniform quorums for mobility management is presented in [22]. A set of nodes in the network form a virtual backbone and each of these nodes stores a location database. The location databases are organized into quorums (sets of databases) in such a way that any two quorums have to intersect by having a shared number of databases. Location updates are sent to any quorum and stored by all its location databases. Due to the intersection between quorums, queries for a node location sent to any quorum should reach a database that maintains a location for that node.

General geographic rendezvous mechanisms could also be used for node location. In [54], we have presented a geographic-based rendezvous architecture, *Rendezvous Regions*, which could be adjusted to provide a node location service. In Rendezvous Regions the network topology is divided into geographical regions. For a node location service, each region will be responsible for a set of nodes. Based on a hash-table-like mapping scheme, each node ID will be mapped to a region. Each node will store its location in the corresponding region and other nodes looking for its location could retrieve it from there. Inside each region, a few elected nodes are responsible for maintaining the information of the mapped nodes. The evaluations have shown that Rendezvous Regions is scalable, efficient and robust to node mobility, failures and location inaccuracy. In Section 4, we will explain Rendezvous Regions in more detail.

2.3 Location Inaccuracy

Geographic routing protocols typically assumed the availability of accurate location information which is necessary for their correct operation. However, in all localization systems an estimation error is incurred that depends on the system and the environment in which it is used. GPS is relatively accurate, but it requires visibility to its satellites and so is ineffective indoors or under coverage. In addition, the high cost, size, and power requirements make it impractical to deploy GPS on all nodes. Infrastructure-based localization systems [63][50] are mostly designed to work inside buildings and they either have a coarse-granularity of several meters or require a costly infrastructure. In ad hoc localization systems [11][53], nodes calculate their locations based on measurements to their neighbors or to other reference nodes in the environment. High localization errors can occur due to environmental factors affecting the location measurements such as obstacles. In addition, errors in a node location propagate to other nodes using it as a reference.

In [23], simulation results were shown for the effect of localization errors on the performance of greedy forwarding. The conclusion was that routing performance is not significantly affected when the error is less than 40% of the radio range. Face routing is not considered in that work. In [30], it is assumed that the system can deal with location errors, since a route discovery protocol is used when greedy forwarding fails. If a flooding route discovery approach is used, it will not be affected by the location errors at the cost of high discovery overhead, but if a route discovery approach based on location (e.g. face routing) is used, the route discovery itself could fail. Approaches that use the location for remote routing only and topology-based routing for local routing such as Terminodes [7] can tolerate inaccuracy in the destination location, but inaccuracy in intermediate nodes can still cause failures.

2.3.1 The Effect of Location Inaccuracy on Face Routing

As we mentioned, greedy forwarding coupled with face routing is an efficient approach that guarantees delivery and accordingly it is the most accepted approach among the currently proposed geographic protocols. In the absence of location errors it has been shown to work correctly and efficiently. In [56], we provided a detailed analysis on the effect of location errors on complete geographic routing protocols consisting of greedy forwarding coupled with face routing. The methodology for this analysis is novel: using an elaborate, micro-level analysis of face routing protocols, we provided detailed scenarios in which the protocol correctness is violated when the location of a node is in error. We performed detailed analysis based on the protocol components to classify the errors and specify their conditions and bounds. We also performed extensive simulations to evaluate and quantify the effects of localization errors on a geographic routing protocol and a geographic-based rendezvous mechanism. Based on our analysis and error classification we introduced a simple and elegant protocol fix that eliminates the most likely protocol errors and we evaluated the efficacy of our fix. Our simulations show near perfect performance for our modified geographic routing even in the presence of significant localization errors. This is the first work to point the different pathologies that can happen in planarization due to the violation of the unit graph assumption.

In other studies, we have shown also how location errors, caused by inconsistency of location dissemination [34] or node mobility [58] result in severe performance degradation and correctness problems in geographic routing protocols.

2.4 The Effect of Link Losses on Geographic Routing

In [57], we provided energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. Experimental studies have shown that wireless links in real sensor networks can be extremely unreliable, deviating to a large extent from the idealized perfect-reception-within-range models used in most common network simulation tools. The previous discussed protocols commonly employ a maximum-distance greedy forwarding technique that works well in ideal conditions. However, such a forwarding technique performs poorly in realistic conditions as it tends to forward packets on lossy links. Based on a realistic link loss model, we studied the distance-hop tradeoff via mathematical analysis and extensive simulations of a wide array of blacklisting/link-selection strategies; we also validated some strategies using real experiments on motes. Our analysis, simulations and experiments, all show that the product of the packet reception rate (PRR) and the distance traversed towards destination is a very effective metric with and without ARQ. Nodes using this metric often take advantage of neighbors in the reception transitional region (high-variance links). Our results also show that reception-based strategies are in general better than distance-based, and we also provide blacklisting strategies that reduce the risk of routing disconnections.

In greedy forwarding, each node forwards a packet to the neighbor that is closest to the destination. The link quality of that neighbor may be very bad. The existence of such unreliable links exposes a key weakness in maximum-distance greedy forwarding that we refer to as the *weakest link* problem. At each step, the neighbors that are closest to the destination (also likely to be farthest from the forwarding node) may have poor links with the current node. These “weak links" would result in a high rate of packet drops, resulting in drastic reduction of delivery rate or increased energy wastage if retransmissions are employed. This observation brings to the fore the concept of neighbor classification based on link reliability. Some neighbors may be more favorable to choose than others, not only based on distance, but also based on loss characteristics. This suggests that a blacklisting/neighbor selection scheme may be needed to avoid `weak links'. In [57], we present and study in detail several blacklisting and neighbor selection schemes.

3. Geocasting

Geocasting is the delivery of packets to nodes within a certain geographic area. Perhaps the simplest way for geocasting is global flooding. In global flooding, the sender broadcasts the packet to its neighbors, and each neighbor that has not received the packet before broadcasts it to its neighbor, and so on, until the packet is received by all reachable nodes including the geocast region nodes. It is simple but has a very high overhead and is not scalable to large networks.

Imielinski and Navas [28][44] presented geocasting for the Internet by integrating geographic coordinates into IP and sending the packet to all nodes within a geographic area. They presented a hierarchy of geographically-aware routers that can route packets geographically and use IP tunnels to route through areas not supporting geographic routing. Each router covers a certain geographic area called a service area. When a router receives a packet with a geocast region within its service area, it forwards the packet to its children nodes (routers or hosts) that cover or are within this geocast region. If the geocast region does not intersect with the router service area, the router forwards the packet to its parent. If the geocast region and the service area intersect, the router forwards to its children that cover the intersected part and also to its parent.

Ko and Vaidya [36] proposed geocasting algorithms to reduce the overhead, compared to global flooding, by restricting the forwarding zone for geocast packets. Nodes within the forwarding zone forward the geocast packet by broadcasting it to their neighbors and nodes

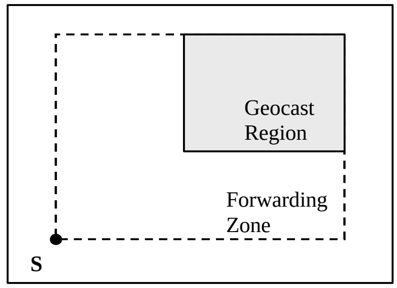

Figure 6: Fixed Rectangular Forwarding Zone (FRFZ)

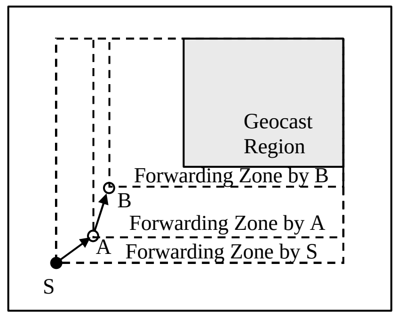

Figure 7: Adaptive Rectangular Forwarding Zone (ARFZ)

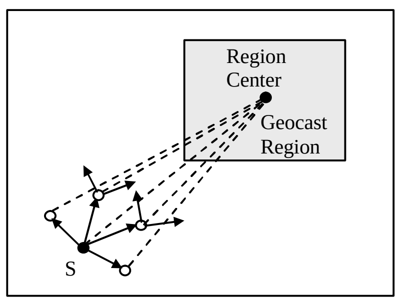

Figure 8: Progressively Closer Nodes (PCN): Closer nodes to the region than the forwarding node forward the packet further and other nodes discard it

outside the forwarding zone discard it. Each node has a localization mechanism to detect its location and to decide when it receives a packet, whether it is in the forwarding zone or not. The algorithms are the following:

- Fixed Rectangular Forwarding Zone (FRFZ) (Figure 6): The forwarding zone is the smallest rectangle that includes the sender and the geocast region. Nodes inside the forwarding zone forward the packet to all neighbors and nodes outside the zone discard it.

- Adaptive Rectangular Forwarding Zone (ARFZ) (Figure 7): Intermediate nodes adapt the forwarding zone to be the smallest rectangle including the intermediate node and the geocast region. The forwarding zones observed by different nodes can be different depending on the intermediate node from which a node receives the geocast packet.

- Progressively Closer Nodes (PCN) (Figure 8): When node B receives a packet from node A, it forwards the packet to its neighbors only if it is closer to the geocast region (center of region) than A or if it is inside the geocast region. Notice that this is different from geographic forwarding; in geographic forwarding a node forwards the packet to the neighbor closest to the region while here a node forwards the packet to all neighbors and *all* neighbors closer to the region forward it further.

Other variations of the FRFZ, ARFZ and PCN mechanisms could also be used, for example by increasing the area of the forwarding zone to include more nodes around the geocast region. These variations could improve the delivery rate at the expense of higher overhead, but they do not provide guaranteed delivery. To reduce the overhead further, GeoTORA [37] uses a unicast routing protocol (TORA [47]) to deliver the packet to the region and then floods within the region. In [59], the network is partitioned using the Voronoi diagram concept and each node forwards the packet to the neighbors whose Voronoi partitions (as seen locally by that node) intersect with the geocast region. The idea is to forward to a neighbor only if it is the closest neighbor to a point in the region.

Variations of global flooding and restricted flooding were presented that use some form of clustering or network divisions to divide the nodes [2][43], such that a single node only in each cluster or division needs to participate in the flooding. This approach can reduce the geocasting overhead by avoiding unnecessary flooding to all nodes at the cost of building and maintaining the clusters. Some approaches (e.g. mesh-based) [8][12] use flooding or restricted flooding only initially, to discover paths to nodes in the geocast region, then these paths are used to forward the packets.

Bose *et al.* [9] presented graph algorithms for extracting planar graphs and for face routing in the planar graphs to guarantee delivery for unicasting, broadcasting, and geocasting. For geocasting they provided an algorithm for enumerating all faces, edges, and vertices of a connected planar graph intersecting a region. The algorithm is a depth-first traversal of the face tree and works by defining a total order on the edges of the graph and traversing these edges. An entry edge, where a new face in the tree is entered, needs to be defined for each face based on a certain criteria. In order to determine the entry edges of faces using only local information and without a preprocessing phase, at each edge the other face containing the edge will need to be traversed to compare its edges with the current edge. This could lead to very high overhead.

3.1 Efficient Geocasting Protocols with Perfect Delivery

In [55], we presented efficient and practical geocasting protocols that combine geographic routing mechanisms with region flooding to achieve high delivery rate and low overhead. The challenging problem in geocasting is distributing the packets to all the nodes within the geocast region with high probability but with low overhead. According to our study we noticed a clear tradeoff between the proportion of nodes in the geocast region that receive the packet and the overhead incurred by the geocast packet especially at low densities and irregular distributions. We presented two novel protocols for geocasting that achieve high delivery rate and low overhead by utilizing the local location information of nodes to combine geographic routing mechanisms with region flooding. We have shown that the first protocol, Geographic-Forwarding-Geocast (GFG), has close-to-minimum overhead in dense networks and that the second protocol, Geographic-Forwarding-Perimeter-Geocast (GFPG), provides guaranteed delivery without global flooding or global network information even at low densities and with the existence of region gaps. GFPG is based on the observation that by traversing all faces intersecting a region in a connected planar graph, every node of the graph inside the region is traversed. Our algorithm is efficient by using a combination of face routing and region flooding and initiating the face routing only at specific nodes. In the following section, we explain these protocols in more detail.

3.1.1 Geographic-Forwarding-Geocast (GFG)

In geocast applications, nodes are expected to be aware of their geographic locations. Geographic-Forwarding-Geocast utilizes this geographic information to forward packets efficiently toward the geocast region. A geographic routing protocol consisting of greedy forwarding with perimeter (face) routing is used by nodes outside the region to guarantee the forwarding of the packet to the region. Nodes inside the region broadcast the packet to flood the region. An example is shown in Figure 9. In more detail, a node wishing to send a geocast creates a packet and puts the coordinates of the region in the packet header. Then it forwards the packet to the neighbor closest to the destination. The destination of geographic routing in this case is the region center. Each node successively forwards the packet to the neighbor closest to the destination using greedy forwarding. When greedy forwarding fails, perimeter routing is used to route around dead-ends until closer nodes to the destination are found. Ultimately (in case there are nodes inside the region) the packet will enter the region. The first node to receive the geocast packet inside the region starts flooding the region by broadcasting to all neighbors. Each node inside the region that receives the packet for the first time broadcasts it to its neighbors and nodes outside the region discard the packet. For region flooding, smart flooding approaches [45] could also be used to reduce the overhead.

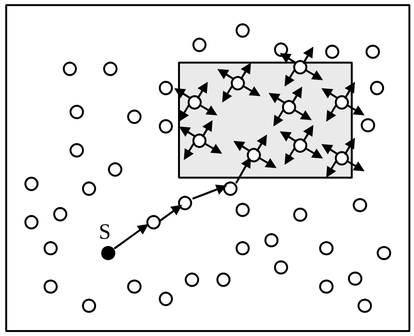

Figure 9: Sender S sends a geocast packet, geographic forwarding is used to deliver the packet to the region, then it is flooded in the region

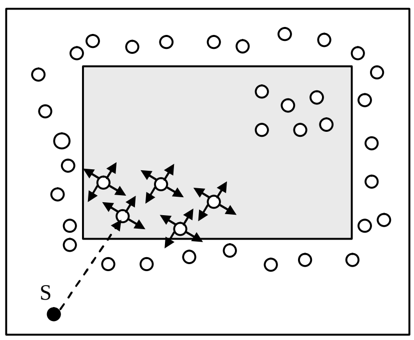

Figure 10: A gap (disconnection) in the geocast region. A packet flooded in the region cannot reach all nodes without going out of the region

In dense networks without obstacles or gaps, GFG is sufficient to deliver the packet to all nodes in the region. In addition, since in dense networks geographic routes are close to optimal routes (shortest path), GFG has almost the minimum overhead a geocast algorithm can have which mainly consists of the lowest number of hops to reach the region plus the number of nodes inside the region itself.

In order for GFG to provide perfect delivery (i.e. all nodes in the region receive the geocast packet), the nodes in the region need to be connected together such that each node can reach all other nodes without going out of the region. In dense networks normally this requirement is satisfied, but in sparse networks or due to obstacles, regions may have gaps such that a path between two nodes inside the region may have to go through other nodes outside the region as shown in Figure 10. In case of region gaps, GFG will fail to provide perfect delivery. GFPG overcomes this limitation.

3.1.2 Geographic-Forwarding-Perimeter-Geocast (GFPG)

We present an algorithm that guarantees the delivery of a geocast packet to all nodes inside the geocast region, given that the network as a whole is connected. The algorithm solves the region gap problem in sparse networks, but it causes unnecessary overhead in dense networks. Therefore, we present another practical version of the algorithm that provides perfect delivery at all densities and keeps the overhead low in dense networks. The practical version is not guaranteed as the original version, but the simulation results show that practically it still achieves perfect delivery.

This algorithm uses a mix of geocast and perimeter routing to guarantee the delivery of the geocast packet to all nodes in the region. To illustrate the idea, assume there is a gap between two clusters of nodes inside the region. The nodes around the gap are part of the same planar face. Thus if a packet is sent in perimeter mode by a node on the gap border, it will go around the gap and traverse the nodes on the other side of the gap (see Figure 11). The idea is to use perimeter routing on the faces intersecting the region border in addition to flooding inside the region to reach all nodes. In geographic face routing protocols as GPSR, a planarization algorithm is used to create a planar graph for perimeter routing. Each node runs the planarization algorithm locally to choose the links (neighbors) used for perimeter forwarding. The region is composed of a set of planar faces with some faces totally in the region and other faces intersecting the borders of region. Traversing all faces guarantees reaching all nodes in the region.

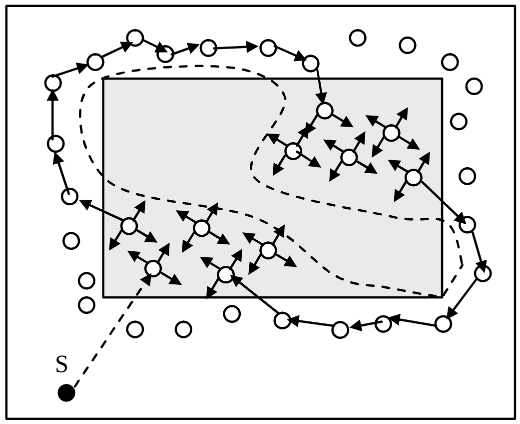

Figure 11: A mix of region flooding and face routing to reach all nodes in the region. Nodes around the gap are part of the same face. For clarity, here we are showing only the perimeter packet sent around the empty face, but notice that all region border nodes will send perimeter packets to their neighbors that are outside of the region

We describe now the algorithm in more detail; please refer to Figure 11. Initially, similar to GFG, nodes outside of the geocast region use geographic forwarding to forward the packet toward the region. As the packet enters the region, nodes flood it inside the region. All nodes in the region broadcast the packet to their neighbors similar to GFG, in addition, all nodes on the border of the region send perimeter mode packets to their neighbors that are outside of the region. A node is a region border node if it has neighbors outside of the region. By sending perimeter packets to neighbors outside the region (notice that perimeter mode packets are sent only to neighbors in the planar graph not to all physical neighbors), the faces intersecting the region border are traversed. The node outside the region, receiving the perimeter mode packet, forwards the packet using the right-hand rule to its neighbor in the planar graph and that neighbor forwards it to its neighbor and so on. The packet goes around the face until it enters the region again. The first node inside the region to receive the perimeter packet floods it inside the region or ignores it if that packet was already received and flooded before. Notice that all the region border nodes send the perimeter mode packets to their neighbors outside of the region, the first time they receive the packet, whether they receive it through flooding, face routing, or the initial geographic forwarding. This way if the region consists of separated clusters of nodes, a geocast packet will start at one cluster, perimeter routes will connect these clusters together through nodes outside the region, and each cluster will be flooded as the geocast packet enters it for the first time. This guarantees that all nodes in the region receive the packet, since perimeter packets going out of the region will have to enter the region again from the opposite side of the face and accordingly all faces intersecting the region will be covered.

Due to the perimeter traversals of faces intersecting the region, GFPG will cause additional overhead that may not be required especially in dense networks, where as we mentioned GFG has optimal overhead by delivering the packet just to nodes inside the region. Ideally we would like perimeter routes to be used only when there are gaps inside the region such that we have perfect delivery also in sparse networks and minimum overhead in dense networks. In this section, we present an adaptation for the algorithm, in which perimeter packets are sent only when there is a suspicion that a gap exists. This new algorithm GFPG*, as the simulations show, practically has perfect delivery in all scenarios. In this algorithm each node inside the geocast region divides its radio range into four portions as shown in Figure 12(a) and determines the neighbors in each portion. This can be done easily, since each node knows its own location and its neighbors' locations. If a node has at least one neighbor in each portion, it will assume that there is no gap around it, since its neighbors are covering the space beyond its range and so it will not send a perimeter packet and will send only the region flood by broadcasting to its neighbors. If a node has no neighbors in a portion, then it sends a perimeter mode packet using the right-hand rule to the first neighbor counterclockwise from the empty portion as shown in

Figure 12(b). Thus the face around the suspected void will be traversed and the nodes on the other side of the void will receive the packet. Notice that in this algorithm there is no specific role for region border nodes and that perimeter packets can be sent by any node in the region, since the gap can exist and need to be detected anywhere. Therefore there are two types of packets in the region, flood packets and perimeter packets. Nodes have to forward perimeter packets even if that packet was flooded before. If a node receives a perimeter packet from the same neighbor for the second time, the packet is discarded, since this means that the corresponding face is already traversed. A node may receive the perimeter packet from different neighbors and thus forwards it on different faces. Our results show the improvement achieved by GFPG* in reducing the overhead at high densities. GFPG* does not guarantee delivery as GFPG, but our simulation results show that practically it has perfect delivery at all densities, in addition to close-to-minimum overhead at high densities. This is desirable for many types of high density applications. The evaluation results of our protocols are in [55].

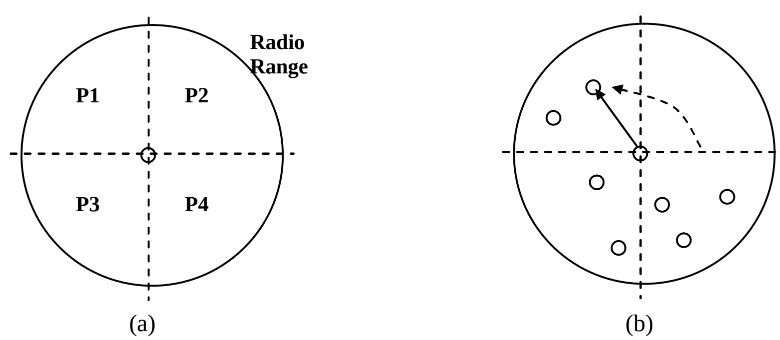

Figure 12: (a) A node divides its radio range into four portions
(b) If a node has no neighbors in a portion, it sends a perimeter packet using the right-hand rule to the first node counterclockwise from the empty portion

4. Geographic-based Rendezvous

In geographic-based rendezvous mechanisms, a geographical location is used as a rendezvous place for providers and seekers of information. Geographic-based rendezvous mechanisms can be used as an efficient means for service location and resource discovery in ad hoc networks. They can also provide efficient data dissemination and access in sensor networks.

In wireless networks, the simplest form of data dissemination and resource discovery is global flooding. This scheme does not scale well. Other approaches that address scalability employ hierarchical schemes based on cluster-heads or landmarks [40]. These architectures, however, require complex coordination between nodes, and are susceptible to major re-configuration (e.g., adoption, re-election schemes) due to mobility or failure of the cluster-head or landmark, incurring significant overhead. GLS [41] provides a scalable location service by using a predefined geographic hierarchy and a predefined ordering of node identifiers to map nodes to their locations. GLS is presented for locating nodes and assumes that node identifiers are known. It is not clear that GLS could be extended efficiently to provide a general rendezvous-based mechanism. One way is to map keys to node identifiers and let the insertion and lookup use GLS to reach that node for storage and retrieval, respectively. A problem here is how nodes can guarantee that a node with that identifier exists and how to do reliable replication at multiple nodes. In addition, the path will be significantly longer, since the insertion or lookup has to find a location server first to get the node location and then it goes to the storage node. Another possibility is to use the key identifier itself to perform storage of the key-value pair in GLS servers similar to how node locations are stored. Since, in GLS the servers of a node are determined based on the node's location, the servers of a key will be determined based on the inserter location. This will create inconsistencies if multiple nodes can insert the same key.

Recently, some geographic-based rendezvous mechanisms have been proposed for data-centric storage in sensor networks. GHT [51] is a geographic hash table system that hashes keys into geographic *points*, and stores the key-value pair at the sensor node closest to the hash of its key.

GHT requires nodes to know their exact geographic location and uses geographic routing to reach the destination. It uses GPSR [33] for geographic routing, where it uses GPSR perimeter routing in a novel way to identify a packet home node (the node closest to the geographic destination). Packets enter perimeter mode at the home node (since no neighbor could be closer to the destination), and traverse the perimeter that encloses the destination (home perimeter) before returning back to home node. GHT uses a perimeter refresh protocol to replicate keys at nodes in the home perimeter. The perimeter refresh protocol refreshes keys periodically using also perimeter routing to deal with topology changes after failures or mobility. ARI [69] is another geographic-based rendezvous scheme for data-centric storage in sensor networks. In this scheme data are stored at nodes close to detecting nodes, and the location information of these storing nodes is pushed to some index nodes. The index nodes for a certain event type form a ring around a rendezvous location for that type. The idea of this scheme is that the nodes in the index ring capture storage and query messages passing through the ring. In order for the index nodes to do that, GAF [65] is used to divide the network into grids with a single node in each grid responsible for forwarding messages. Since GAF is based on the assumption that each node can only forward messages to the nodes in its neighboring grids, a message sent by a node outside of the ring-encircled region and destined to the index center, must pass some nodes on the ring. There are other variations of schemes providing data-centric storage in sensor networks. DIFS [20] is a system built on top of GHT to provide range searches for event properties in sensor networks. Another system, DIM [42] allows multidimensional range queries in sensor networks, which is useful in correlating multiple events. DIM uses a geographic embedding of an index data structure (multidimensional search tree) to provide a geographic locality-preserving hash that maps a multi-attribute event to a geographic zone. The sensor field is logically divided into zones such that there is a single node in each zone. DIMENSIONS [19] provides multi-resolution storage in sensor networks by using wavelet summarization and progressive aging of the summaries in order to efficiently utilize the network storage capacity.

In [3][61], geographic curves are used for match-making between producers and consumers of content. The idea is for producers to send their advertisements along the four directions (north, south, east, and west) and for consumers to send queries also along the four directions. Nodes where advertisements and queries intersect will reply back to the consumers.

4.1 Rendezvous Regions

In [54], we provided a scalable rendezvous-based architecture for wireless networks, called Rendezvous Regions (RR). The original RR idea borrowed from our earlier work on PIM-SM rendezvous mechanism [15] that uses consistent mapping to locate the rendezvous point (RP). However, a rendezvous *point* is insufficient in a highly dynamic environment as wireless networks. We first hinted at the RR idea in [26], in the context of bootstrapping multicast routing in large-scale ad hoc networks, with no protocol details or evaluations. In [54], we presented the detailed architecture for RR, with full description of the design and the mechanisms to deal with mobility, failures, and inaccuracies, and generalizing it to deal with resource discovery and data-centric architectures in general. A main goal in RR design is to target high mobility environments and this makes rendezvous regions more suitable than rendezvous points. RR is also based on our objective to design geographic systems that need only approximate location information. The use of regions affects many design details such as the server election, insertion, lookup, and replication. In Rendezvous Regions, the network topology space is divided into rectangular geographical regions, where each region is responsible for a set of keys representing the data or resources of interest. A key, k_i, is mapped to a region, RR_j, by using a hash-table-like mapping function, $h(k_i)=RR_j$. The mapping is known to all nodes and is used during the insertion and lookup operations. A node wishing to insert or lookup a key obtains the region responsible for that key through the mapping, then uses geographic-aided routing to send a message to the region. Inside a region, a simple local election mechanism dynamically promotes nodes to be servers responsible for maintaining the mapped information. Replication between servers in the region reduces the effects of failures and mobility. By using regions instead of points, our scheme requires only approximate location information and accordingly is more robust to errors and imprecision in location measurement and estimation than schemes depending on exact location information. Regions also provide a dampening factor in reducing the effects of mobility, since no server changes are required as long as current servers move inside their region and hence the overhead due to mobility updates is quite manageable.

The network topology space is divided into geographical regions (*RRs*), where each region (e.g., RR_j) is responsible for a set of resources. The resource key space is divided among these regions, such that each resource key (K_i) is mapped to a region. The key-set to *RR* mapping ($KSet_i \leftrightarrow RR_j$) is known by all nodes.

The Rendezvous Regions scheme can be built on top of any routing protocol that can route packets toward geographic regions. The only requirement of the routing protocol is to maintain approximate geographic information, such that given an insertion or lookup to a certain region, it should be able to obtain enough information to route the packet toward that region. In our design we use geocasts for insertions and anycasts for lookups. These design choices are simple to implement, robust to dynamics, and do not require tracking of nodes' locations. Following we describe the main components of our architecture.

Region detection: Using a localization mechanism, each node detects its location and accordingly its geographic region. When the node moves, it detects its new location and so it can keep track of its region. The node uses this information to forward packets toward regions, to detect packets forwarded to its region, and to potentially participate in server election in its region (if and when needed).

Server election: A simple local election mechanism is used inside the region to dynamically promote the servers. As the number of servers increases, the robustness to mobility and failures increases, but also the storage overhead increases. Servers are elected on-demand during insertions. When a data insertion operation is issued, the first node in the region that receives the insertion[3], known as the *flooder*, geocasts the insertion inside the region. Each server receiving the insertion geocast sends an Ack back to the flooder. The flooder keeps track of the servers and if it does not get enough Acks (the minimum number of servers required), it geocasts again and includes a self-election probability, *p*, in the goecast message. Each node receiving the geocast elects itself with probability *p* and if it becomes a server, it replies to the flooder. If not enough Acks are received, the flooder increases *p* based on a back-off mechanism until the required number of servers reply or *p* reaches 1. When servers move out of the region or fail, new servers are elected in the same way. After the new servers are elected, they retrieve the stored keys from other servers.

[3] A node can identify that it is the first node in the region to receive the packet by a simple flag set in the packet header.

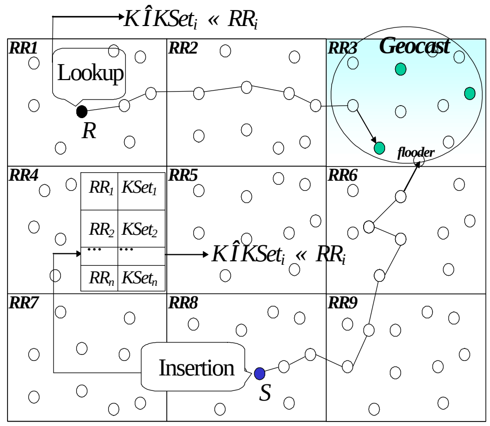

Figure 13: Rendezvous Regions
Insertion: Node *S* wishing to insert (or store) resource key *K* that belongs to $KSet_i$ gets the corresponding *RR* (in this case *RR3*) through the mapping ($KSet_i \rightarrow RR_i$). Node *S* then sends the resource information towards *RR3*, where it is geocast by the flooder and stored by the servers.
Lookup: Node *R* looking for a resource with key *K* that belongs to $KSet_i$ gets the corresponding *RR* (in this case *RR3*) through the mapping ($KSet_i \rightarrow RR_i$). *R* then sends the resource lookup towards *RR3*, where it is anycast to any server holding the information.

Insertion: A node inserts a key, *K*, by first mapping the key to a rendezvous region, RR_i, where $K \in KSet_i \leftrightarrow RR_i$. The node generates a packet containing the region identifier, RR_i, in its header. Nodes routing the packet toward the region, check the region identifier to determine whether they are in or out of region. The first node inside RR_i to receive the packet, the *flooder*, geocasts the packet inside the region. Servers inside the region receive the geocast, store the key and data, then send Acks back to the flooder (Figure 13). The flooder collects the Acks and sends an Ack back to original sender. If no Ack is received by the sender, it timeouts and retransmits the insertion up to a fixed number of times.

Lookup: Lookups are similar to insertions except that nodes and previous flooders inside a region cache locations of the recent servers they hear from, and send the lookups directly to any of the servers (anycast). The server replies to the flooder and the flooder replies back to original sender (Figure 13). If the flooder receives no reply or if it has no cached servers, it geocasts the lookup inside the region.

Replication: Replication is inherent in this architecture, since several servers inside the region store the key and data. This adds extra robustness to failures and mobility. For additional robustness against severe dynamics such as group failures and partitions, multiple hash functions may be used to hash a key to multiple regions.

Mobility: Local movements of nodes and servers have negligible effect and overhead on our architecture as long as servers stay within their regions. The only condition we need to consider is when a server moves out of its region. The server checks its location periodically to detect when it gets out of its region, in order to send an insertion packet toward that region so that new servers are elected. The server then deletes its stored keys and is not a server anymore. It may or may not get elected again later in a new region.

Failures: Since each region contains several servers, and insertions and mobility may invoke new server elections, it is unlikely that independent reasonable failures will cause all servers to vanish. In order to avoid this case anyway, servers use a low-frequency periodic soft-state mechanism during silent (low traffic) periods, to detect failing servers and promote new servers. Each server runs a low-frequency timer, which is reset each time an insertion geocast is received. When the server times out, it geocasts a packet checking for other servers. Other servers reset their timers upon receiving this check and reply back demonstrating their existence. If not enough servers reply back, server election is triggered.

Bootstrap: One question remaining is how the mapping function is obtained. One option is to assume that it is pre-known or provided by out-of-band mechanisms. Another option is to use the same rendezvous mechanism, in order to provide a bootstrap overlay that publishes dynamic mappings. Using the mapping for a *well-known key*, a node sends request to a well-known region to obtain the mapping function of a set of services. These mappings however are not expected to change frequently. This introduces more flexibility for providing different mappings for different type of services and changing them when required.

5. Conclusions

We have presented an overview of geographic protocols for wireless ad hoc and sensor networks. It is obvious that utilizing the geographic information is vital for building scalable and efficient protocols in these environments. This study shows that there is a significant amount of work done in this area. Nevertheless, in order for geographic protocols to be implemented in the real-world, they need a higher degree of robustness to the realistic environmental conditions. In our work, we focus on this issue of assessing the robustness of geographic protocols to non-ideal conditions corresponding to the real-world environments and designing new strategies and protocols that take these conditions into account. We pointed to some of our studies in this paper: the effect of inaccurate locations, lossy wireless links, robust geocasting, and Rendezvous Regions. In the present and future work, we are considering additional issues to improve the robustness of geographic protocols and move them closer to effective real-world deployment.

References

[1] I. Akyildiz, J. Ho, and Y. Lin. "Movement-Based Location Update and Selective Paging for PCS Networks". *IEEE/ACM Transactions on Networking*, August 1996.

[2] B. An and S. Papavassiliou. "Geomulticast: Architectures and Protocols for Mobile Ad Hoc Wireless Networks". *Journal of Parallel and Distributed Computing*, v.63 n.2, p.182-195, February 2003.

[3] I. Aydin, C.-C. Shen. "Facilitating Match-Making Service in Ad hoc and Sensor Networks Using Pseudo Quorum". *IEEE ICCCN* 2002.

[4] L. Barrière, P. Fraigniaud, L. Narayanan, and J. Opatrny. "Robust Position-Based Routing in Wireless Ad Hoc Networks with Irregular Transmission Ranges". Wireless Communications and Mobile Computing 2(3), 141-153, 2003.

[5] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward. "A Distance Routing Effect Algorithm for Mobility (DREAM)". *ACM MOBICOM* 1998.

[6] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. P. Hubaux, and J. Y. Le Boudec. "Self-Organization in Mobile Ad-Hoc Networks: the Approach of Terminodes". *IEEE Communications Magazine*, June 2001.

[7] L. Blazevic, S. Giordano, J.-Y. Le Boudec. "Self Organized Terminode Routing". *Cluster Computing Journal, Vol.5, No.2*, April 2002.

[8] J. Boleng, T. Camp, and V. Tolety. "Mesh-Based Geocast Routing Protocols in an Ad Hoc Network". IPDPS 2001.

[9] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks". *Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM* 1999*)*.

[10] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. “A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols”. *ACM MOBICOM* 1998.

[11] N. Bulusu, J. Heidemann, D. Estrin, and T. Tran. “Self-configuring Localization Systems: Design and Experimental Evaluation”. *ACM Trans on Embedded Computing Systems (TECS)* 2003.

[12] T. Camp and Y. Liu. “An Adaptive Mesh-Based Protocol for Geocast Routing”. Journal on Parallel and Distributed Computing: Special Issue on Routing in Mobile and Wireless Ad Hoc Networks, Feb. 2003, pp. 196–213.

[13] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. “A High-Throughput Path Metric for Multi-Hop Wireless Routing”. ACM MobiCom, September 2003.

[14] D. De Couto and R. Morris. “Location Proxies and Intermediate Node Forwarding for Practical Geographic Forwarding”. *MIT Technical Report*, June 2001.

[15] D. Estrin, M. Handley, A. Helmy, P. Huang, and D. Thaler. “A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing”. *IEEE INFOCOM 1999*.

[16] G. G. Finn. “Routing and Addressing Problems in Large Metropolitan-Scale Internetworks”. *Tech. Rep. ISI/RR-87-180*, Information Sciences Institute, Mar. 1987.

[17] K. Gabriel, R. Sokal. “A New Statistical Approach to Geographic Variation Analysis”. *Systematic Zoology 18* (1969), 259–278.

[18] J. Gao, L. Guibas, J. Hershberger, L. Zhang, A. Zhu. “Geometric Spanners for Routing in Mobile Networks”. *ACM MOBIHOC* 2001.

[19] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann. “An Evaluation of Multi-resolution Storage for Sensor Networks”. ACM SenSys 2003.

[20] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker. “DIFS: A Distributed Index for Features in Sensor Networks”. *IEEE SNPA* 2003.

[21] Z. Haas. “A New Routing Protocol for the Reconfigurable Wireless Networks”. *IEEE Conf. on Universal Personal Comm.*, pp. 562-6, ‘97.

[22] Z. J. Haas and B. Liang. “Ad Hoc Mobility Management with Uniform Quorum Systems”. *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, pp. 228-240, April 1999.

[23] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. “Range-Free Localization Schemes for Large Scale Sensor Networks”. *ACM MOBICOM* 2003.

[24] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. “SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks”. *International Conference on Distributed Computing Systems*, May 2003.

[25] J. Heidemann, F. Silva, and D. Estrin. “Matching Data Dissemination Algorithms to Application Requirements”. *ACM Sensys* 2003.

[26] A. Helmy. “Architectural Framework for Large-Scale Multicast in Mobile Ad Hoc Networks”. *IEEE ICC*, April 2002.

[27] J. Hightower and G. Borriello. “Location Systems for Ubiquitous Computing”. *IEEE Computer*, Aug. 2001.

[28] T. Imielinski and J. Navas. “GPS-based Addressing and Routing”. IETF RFC 2009, Nov. 1996.

[29] C. Intanagonwiwat, R. Govindan, and D. Estrin. “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks”. *ACM MOBICOM* 2000.

[30] R. Jain, A. Puri, and R. Sengupta. “Geographical Routing Using Partial Information for Wireless Ad Hoc Networks”. *IEEE Personal Communications*, Feb 2001.

[31] D. B. Johnson and D. A. Maltz. “Dynamic Source Routing in Ad-Hoc Wireless Networks”. *Mobile Computing* 1996.

[32] B. Karp. “Challenges in Geographic Routing: Sparse Networks, Obstacles, and Traffic Provisioning”. Slides presented at the DIMACS workshop on Pervasive Networking, May 2001.

[33] B. Karp and H.T. Kung. “GPSR: greedy perimeter stateless routing for wireless networks”. *ACM MOBICOM* 2000.

[34] Y. Kim, J. Lee, and A. Helmy. “Impact of Location Inconsistencies on Geographic Routing in Wireless Networks”. *ACM MSWIM* 2003.

[35] Y. Ko and N. Vaidya. “Location-Aided Routing (LAR) in Mobile Ad Hoc Networks”. *ACM MOBICOM* 1998.

[36] Y. Ko and N. Vaidya. “Flooding-based Geocasting Protocols for Mobile Ad Hoc Networks”. *ACM/Baltzer Mobile Networks and Applications (MONET) Journal*, Dec. 2002.

[37] Y. Ko and N. Vaidya. “Anycasting-based Protocol for Geocast service in Mobile Ad Hoc Networks”. *Computer Networks Journal*, Apr. 2003.

[38] E. Kranakis, H. Singh, and J. Urrutia. “Compass Routing on Geometric Networks”. *In Proc. 11th Canadian Conference on Computational Geometry*, August 1999.

[39] F. Kuhn, R. Wattenhofer, and A. Zollinger. “Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing”. *ACM MOBIHOC* 2003.

[40] S. Kumar, C. Alaettinoglu, and D. Estrin. “Scalable Object-tracking through Unattended Techniques (SCOUT)”. *IEEE ICNP* 2000.

[41] J. Li, J. Jannotti, D. Couto, D. Karger, and R. Morris. “A Scalable Location Service for Geographic Ad Hoc Routing”. *ACM Mobicom* 2000.

[42] X. Li, Y. Kim, R. Govindan, and W. Hong. “Multi-Dimensional Range Queries in Sensor Networks”. ACM Sensys, November 2003.

[43] W.H. Liao, Y.C. Tseng, K.L. Lo, and J.P. Sheu. “GeoGRID: A Geocasting Protocol for Mobile Ad Hoc Networks Based on GRID”. *Journal of Internet Technology*, vol. 1, no. 2, pp. 23–32, Dec. 2000.

[44] J. Navas and T. Imielinski. “Geographic Addressing and Routing”. *ACM MOBICOM* 1997.

[45] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. “The Broadcast Storm Problem in a Mobile Ad Hoc Network”. ACM MOBICOM 1999.

[46] D. Nicules and B. Nath. “Trajectory-Based Forwarding and Its Applications”. *ACM MOBICOM* 2003.

[47] V. Park and M. Corson. “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks”. *IEEE INFOCOM* 1997.

[48] C. E. Perkins and P. Bhagwat. “Highly Dynamic Destination-Sequenced Distance Vector Routing for Mobile Computers”. *ACM CCR* Oct. 1994.

[49] C Perkins and E. Royer. “Ad hoc On-Demand Distance Vector Routing”. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.

[50] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. “The Cricket Location-Support System”. *ACM MOBICOM* 2000.

[51] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. “GHT: A Geographic Hash Table for Data-Centric Storage”. *ACM WSNA*, Sept 2002 and ACM MONET 2003.

[52] E. Royer and C. Toh. “A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks”. *IEEE Personal Communications Magazine*, April 1999, pp. 46-55.

[53] A. Savvides, C.-C. Han, and M. B. Srivastava. “Dynamic Fine-Grain Localization in Ad-Hoc Networks of Sensors”. *ACM MOBICOM* 2001.

[54] K. Seada and A. Helmy. “*Rendezvous Regions*: A Scalable Architecture for Service Location and Data-Centric Storage in Large-Scale Wireless Networks”. *IEEE/ACM IPDPS 4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), Santa Fe, New Mexico,* April 2004.

[55] K. Seada, A. Helmy. “Efficient Geocasting with Perfect Delivery in Wireless Networks”. *IEEE Wireless Communications and Networking Conference (WCNC), Atlanta, Georgia,* March 2004.

[56] K. Seada, A. Helmy, and R. Govindan. “On the Effect of Localization Errors on Geographic Face Routing in Sensor Networks”. *IEEE/ACM 3rd International Symposium on Information Processing in Sensor Networks (IPSN), Berkeley, CA,* April 2004.

[57] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari. “Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks”. *ACM SenSys* 2004.

[58] D. Son, A. Helmy, and B. Krishnamachari. “The Effect of Mobility-induced Location Errors on Geographic Routing in Ad Hoc Networks: Analysis and Improvement using Mobility Prediction”. *IEEE WCNC* 2004.

[59] I. Stojmenovic, A. P. Ruhil, and D. K. Lobiyal. “Voronoi Diagram and Convex Hull-Based Geocasting and Routing in Wireless Networks”. IEEE ISCC 2001.

[60] H. Takagi and L. Kleinrock. “Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals”. IEEE Transactions on Communications, Mar. 1984.

[61] J. Tchakarov and N. Vaidya. “Efficient Content Location in Wireless Ad Hoc Networks”. *IEEE International Conference on Mobile Data Management (MDM)*, January 2004.

[62] G. Toussaint. “The Relative Neighborhood Graph of a Finite Planar Set”. *Pattern Recognition 12*, 4 (1980), 261–268.

[63] A. Ward, A. Jones, and A. Hopper. “A New Location Technique for the Active Office”. *IEEE Personal Communications*, October 1997.

[64] S. Woo and S. Singh. “Scalable Routing Protocol for Ad hoc Networks”. Wireless Networks, January 2001.

[65] Y. Xu, J. Heidemann, and D. Estrin. “Geography-informed Energy Conservation for Ad-hoc Routing”. *ACM MOBICOM* 2001.

[66] Y. Xue, B. Li, K. Nahrstedt. “A Scalable Location Management Scheme in Mobile Ad-hoc Networks”. *IEEE Annual Conference on Local Computer Networks (LCN)*, November 2001.

[67] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. “A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks”. *ACM MOBICOM* 2002.

[68] Y. Yu, R. Govindan, and D. Estrin. “Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks”. *USC Technical Report*, 2001.

[69] W. Zhang, G. Cao, and T. La Porta . “Data Dissemination with Ring-Based Index for Sensor Networks”. *IEEE ICNP* 2003.