Network Construction and Routing in Geographic Overlays

Gregory G. Finn, Joseph D. Touch
USC/Information Sciences Institute
July 10, 2002

Abstract—**Geographic networks use physical location as destination addresses, and forward packets topographically. These networks enable geographic and proximity-sensitive applications, unlike the current Internet naming and routing architecture. Geographic routing applied to a native network generally suffers from routing inconsistencies and inefficiencies. A geographic overlay can be incrementally deployed on the current Internet, in ways that ensure routing completeness and consistency. The result is a network with finite, degree 6 routing tables, which are both complete and consistent. The resulting overlay achieves constant-time route calculation, robust unicast and spatialcast forwarding, and supports hierarchical extensions having an expected hop count that is log proportional to network size.**

Index Terms—geographic routing, overlay, tunnel, spatialcast

I. INTRODUCTION

The Internet uses a hierarchical address and routing scheme that reflects customer/provider relationships. Packets are forwarded toward an address, rather than a geophysical location. Hosts may be physically adjacent, but the distance between them could be many hops, interfering with applications that depend on physical location or proximity.

A geographic network, or GNet, would enable new location-based, proximity-based and spatial application classes. The latter two utilize spatialcast, analogous to multicast, but where membership is implicitly defined by location.

Location-based applications depend upon known host location, e.g., reading only those sensors at particular map positions and tracking. Proximity-based applications depend upon discovery of nearby hosts, e.g., to correlate events, or to find resources such as printers or caches. Spatial applications affect hosts in a delimited area, e.g., communicating with planes in a region of airspace, vehicles on a section of road, or evacuating an affected area during an emergency.

Native networks have an interconnection topography defined by their physical, wired or wireless links. Unless the interconnection topography is very carefully constrained, determining which neighbors are physically near can require global knowledge of network topology. Imposing a native GNet onto a network of irregular interconnection topography implies the global exchange of topographic information. Disabled or insufficient links at a node can create geographic routing inconsistencies, which require difficult, global compensation.

A GNet can also be deployed as an overlay network, on top of an existing (e.g., Internet) network. An overlay network is composed of virtual links added to an underlying network interconnection topography. This ability to tailor interconnection topography allows creation of topographies well suited to geographic network requirements. A geographic overlay network (GONet) can thus ensure consistency and completeness.

A GONet, besides providing support for geographic addressing and routing, also enables networks to be incrementally deployed with routing tables that are tightly bounded and small. The ability to recursively build overlays on overlays allows creation of a routing hierarchy in a GONet to achieve typical hop counts that are log proportional to network size.

The remainder of this paper is organized as follows: Section II discusses the gross requirements of geographic unicast routing topographies and introduces a general methodology for achieving a sufficient and complete geographic network. Section III introduces two algorithms, Voronoi and Greedy Edge, that determine suitable interconnection topographies for a GONet. Section IV analyzes the greedy edge algorithm. Recursive application of greedy edge is then discussed as a means to achieve typical hop counts that are log proportional to network size. Section V defines spatialcast and demonstrates that it is compatible with geographic unicast routing topographies. The subject of bootstrapping is also briefly discussed and some open issues are mentioned. Section VI produces confirming results from simulations and Section VII covers related work.

II. GEOGRAPHIC NETWORKING

A GNet routes packets toward geographic locations. Because multiple hosts can be at the same location, a complete address also requires a unique host identifier. Without loss of generality assume an address consists of *<geo, id>* portions, where packets are routed toward the *geo* portion, and the *id* portion is used as a filter. The *geo* part, e.g., may be some combination of latitude, longitude and altitude.

II.1. Geographic Network Graphs

In [1], networks were broken into classes defined by the maximum number of hops necessary to forward any packet closer to its geographic destination. If all routers could forward any packet closer to its destination in *n* or fewer hops, the network was said to be *n-hop consistent.* The following mechanism generates the edges that transform an inconsistent graph into one that is 1-hop consistent.

For the following discussion, assume all hosts are sites on a Euclidean plane; extensions to volumes and compensations for curved surfaces (of the earth) are omitted for simplicity. Also assume that sites are unique; as above, the *id* combined with a filter provides sufficient further resolution.

Consider two sites S_i, S_j and a line segment R between them depicted in Figure 1. The perpendicular bisection of edge R, shown as B, defines two half planes, the left (unshaded) *route-from* half plane, and the right (shaded) *route-to* half plane. S_i can move any packet, whose destination lies within the *route-to* half plane containing S_j, closer to its destination (*make progress*) by forwarding that packet to S_j. The edge R thus represents a forwarding table entry at S_i.
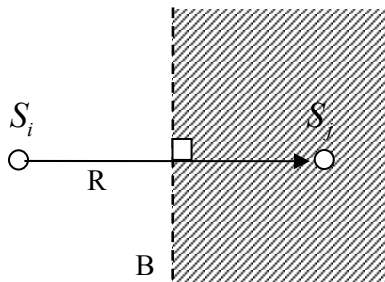

Figure 1 Forward Progress

Consider sets of neighbor sites {**S**} around a router at site S_i, and for each set, consider its effectiveness as a list of forwarding entries at S_i. For some of these sets, the intersection of *route-from* half planes outlines a convex polygonal region that contains only S_i. In that case, S_i can make progress to any location by forwarding to some member of {**S**}. The graph of edges between S_i and {**S**} is 1-hop consistent with respect to S_i. This approach generalizes to three dimensions, and assumes only that the space remains Euclidean.

Applying this procedure to all sites in a network will transform a network graph into a 1-hop consistent GNet. The procedure assumes that any two sites can be connected by an edge. In a native GNet, this requires full, or at least heavy, local connectivity. Incomplete connectivity can cause forwarding dead-ends. If there exists a site that has no set {**S**} producing an appropriate

convex routing polygon, it results in a local dead-end. There is no global way to avoid paths involving such dead ends.

II.2. Virtual Geographic Networking

The links of a geographic network can be created from tunnels in an underlying, existing base network, resulting in a geographic overlay network. The sites in a GONet rely on the base network for link connectivity and can avoid generating inconsistent routing tables. Paths in the base network may take a large number of hops to reach a geographically nearby neighbor, however. Although efficiency is compromised, it produces the desired geographic network topography, which allows unicast geographic routing and spatialcasting.

Methods to discover physically close neighbors in a network and the edges needed to produce a consistent geographic network were studied in [1]. The new edges corresponded to source routes from one to *n* hops long in a base network. The flooding methodology developed there did not scale to large networks. The use of tunnels greatly simplifies that procedure, because source routes need not be discovered nor maintained. Only edge endpoints need be known. Maintaining the route between endpoints becomes the responsibility of the underlying network.

Hosts in a geographic overlay require distinct addresses in both base and overlay networks, with edges to neighbors in the overlay implemented as tunnels in the base. Overlays were developed to deploy new protocols, most notably the M-Bone for multicast IPv4, and have been extended as a general mechanism for network virtualization known as the X-Bone [21]. An X-Bone overlay includes both end hosts and routers in the overlay, and includes support for overlays at all points in the virtual network. A GONet is an X-Bone style overlay where the virtual network is a geographic network.

Because tunnels encapsulate overlay payloads in base packets, the base is isolated from overlay packet formats. Overlay packets are exposed only at tunnel endpoints, where the appropriate overlay stack is implemented. Thus, the overlay may support a differing form of host address and protocol stack from that in the base. In particular, it may implement a geographic address. Not all base components need participate in the overlay; those that do not, are expected to at least forward encapsulated overlay packets.

III. COMPUTING EDGE SETS

There are various methods for computing specific edge sets needed to create a 1-hop consistent network graph.

In a GNet, these are the equivalent of the routing protocol. Various sets of nearby neighbors are discovered and considered. Any set whose *route-from* half planes outline a convex polygon around the router site that contains no other sites is sufficient as a table for that site. Note that computing an edge set may itself use the GNet, as nearby neighbor location requires a proximity discovery mechanism.

There are two algorithms for computing edge sets. The first is a derived from Computational Geometry, and the second was developed to address problems with the first.

III.1. Voronoi Edge Generation

The description above of the solution space for 1-hop consistent graphs is based on the intersection of half-planes around each site. This is strikingly similar to the partition method that defines Voronoi graphs [2][3]. The Voronoi region around a site contains all the points that are closer to that site than to any other site. For each point, take the perpendicular to the bisector as in Figure 1; the intersection of these perpendiculars forms a polygon (Figure 2). Voronoi regions necessarily partition a plane into convex polygons.
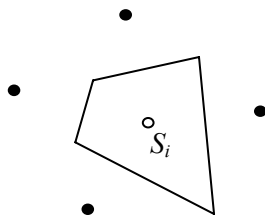

Figure 2 Voronoi Polygon for S_i

By placing an edge between any two sites whose Voronoi polygons share an edge, the space can be triangulated, producing a Delaunay graph (Figure 3). If each site is assigned a geographic address, it can be seen that a Delaunay graph is a 1-hop consistent network. The corresponding routing table for each site would have an entry for each edge of its Voronoi polygon.
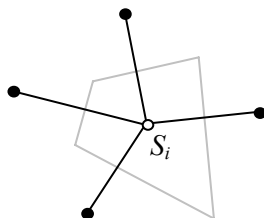

Figure 3 Delaunay Graph for S_i

There are two difficulties in using Voronoi polygons to define the interconnection graph of a GNet. First, Voronoi polygons are not straightforward to calculate in two dimensions and are even more difficult to calculate in three dimensions. Second, the upper bound on the

number of edges of a Voronoi polygon is *N-1*, where *N* is the number of sites, an example of which is a spiral (Figure 4). Thus, if the Delaunay triangulation is used to define site routing tables, some tables may be large.
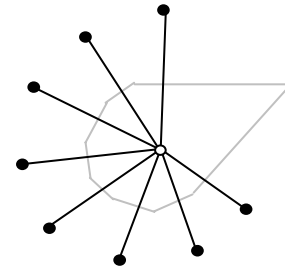

Figure 4 Delaunay Graph for Spiral Pattern

Routing tables are more efficient when bounded. It is thus preferable to find an algorithm for generating edge sets that produces a similar convex polygon but that has a small upper bound on the number of its vertices, regardless of site location patterns. It would also be desirable if that algorithm were both simple and easily extended to three dimensions.

III.2. Greedy Edge Generating Algorithm

Consider a population of sites on a plane and assign them geographic addresses. Consider now a site *H* and the non-empty set {**S**} of its neighbor sites. From {**S**} choose the site closest to *H*, call it N_1. Place N_1 into neighbor set {**N**} and place an edge from *H* to N_1. The bisection of that edge defines half planes. For any destination that lies in the half plane containing N_1, *H* can make progress by forwarding to N_1. Call that the covered region about *H* defined by N_1. The half plane that contains *H* and the line of bisection constitutes an uncovered region, U_1, for which *H* cannot make progress by forwarding to N_1. Discard from {**S**} all sites that lie in the covered region. Now iterate. The algorithm halts when {**S**} is empty, which implies that, with respect to {**S**}, there are no sites within *H*'s uncovered region but *H* itself.
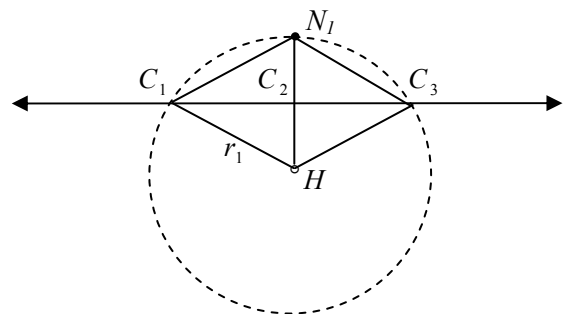

Figure 5 Geometry Imposed by Edge

The intersection of U_i forms a convex *routing polygon* around *H*. Each edge defined by {**N**} determines an entry in *H*'s routing table. If the routing

polygon is closed, those entries constitute a complete routing table for *H*, since no site other than *H* resides in that polygon and all other sites from the population reside within a region covered by some N_i. If the routing polygon is not closed, either that polygon contains no sites other than *H* or another set {**S**} is needed. Unlike Voronoi polygons, the convex polygons generated by this algorithm can overlap.

IV. CHARACTERIZING GREEDY EDGE ALGORITHM

Several practical issues remain to be discussed regarding the greedy edge generating algorithm and the networks that result from its use. What is the upper bound on the number of edges that the polygons generated by the algorithm have and so therefore the number of entries in routing tables? How many hops will the typical path have? How might a host *H* find its geographic neighbors and their addresses?

IV.1. Routing Table Bounds

Consider a population of hosts {**S**} as sites on a plane. Identify one of those sites as *H*. Determine a closest site in {**S**} to *H* at distance r_1. Call that site N_1. The bisection of the line segment $\overline{HN_1}$ defines two half-planes. Any site within the half-plane that contains N_1 is closer to N_1 than to H.

See Figure 5. The bisector of the segment $\overline{HN_1}$ also defines a chord in the circle centered at *H* of radius r_1. The segments $\overline{C_2H}$ and $\overline{C_2N_1}$ are of length $r_1/2$, right angles are formed by C_1C_2H and $C_1C_2N_1$, thus, triangles ΔC_1C_2H and $\Delta C_1C_2N_1$ are congruent. Because $\overline{HC_1}$ is of length r_1, so is $\overline{N_1C_1}$ and ΔHC_1N_1 is equilateral. By symmetry, so is ΔHN_1C_3.

Thus, the angle C_1HC_3 sweeps is 120 degrees. Since no hosts in {**S**} other than *H* lie within the circle defined by *H* and r_1, N_1 covers the interior of the 120 degree region defined by C_1HC_3 of *H*'s routing space. Note that C_1 and C_3 are limit points of open line segments not covered. Thus, each site provides up to 120 degrees of coverage.

Discard from {**S**} all sites that lie in the N_1 half-plane and presume {**S**} not empty. Choose the next closest site in {**S**} to *H* at distance r_2. Call that site N_2. The distance $r_2 \geq r_1$. Without loss of generality, assume N_2 is closer to C_1 than C_3. As above, bisect the line segment HN_2. The minimum portion of *H*'s routing space covered by N_2 occurs when $N_2 = C_1$ and $r_2 = r_1$. By reference to the above geometry, N_2 covers an additional 60 degrees of *H*'s routing space, while it overlaps the 60-degree region $C_1H N_1$. Thus, a second site added to *H*'s routing table must cover at least an additional 60 degrees but no more than 120 degrees.

Clearly, three sites could be situated on a circle around H, separated from one another by 120 degrees. In that case three limit points would remain uncovered. Include those limit points in H's routing table and a total of six entries results.

Since the first site covers 120 degrees and the next covers at least an additional 60 degrees, five sites will cover at least 360 degrees. However, in the worst case if all $N_2...N_5$ cover 60 additional degrees, one open limit point remains uncovered. Include that point in *H's* routing table and a total of six entries results.

In two dimensions, the greedy algorithm produces a lower bound on a closed routing polygon of three entries and an upper bound of six entries, five in practice. Thus, in two dimensions packet routing may be done in constant time. A proof for the upper bound in three dimensions was not discovered. From extensive simulation it appears that it is eight in practice.

IV.2. Typical Hop Count

Assume network edges are defined by the greedy edge algorithm and that the route chosen at each hop is that which makes best progress toward the destination. If the number of routers is large and their locations are uniformly randomly distributed, end-to-end paths will tend toward a straight line to the destination. The expected hop count is then proportional to distance to the destination divided by mean hop length. The diameter of a two-dimensional network would grow at a rate that is proportional to the square root of the number of routers and for three-dimensional networks at a rate proportional to the cube root.

While growth at that rate is suitable for small to moderate size networks, for large networks a growth rate that is log proportional is desired. The freedom to add edges to an overlay allows the augmenting of its network graph to decrease diameter. It is possible to achieve a log proportional diameter with a small increase in route table size.

IV.3. Recursive Greedy Edge Algorithm

The greedy edge algorithm can be applied recursively to create a geophysical routing hierarchy. Consider a lowest $level_0$ network organized by the greedy edge algorithm. From among the $level_0$ routers randomly choose some proportion $1/P$ to be members of a $level_1$ hierarchy. Use the greedy edge algorithm to produce a network graph and route tables for those $level_1$ routers. $Level_1$ routers then have routes to both $level_0$ and $level_1$ neighbors.

Determine for each $level_0$ host L_0 the closest $level_1$ node L_1. This can be obtained via a $level_0$ spatialcast. L_0 adds an edge to L_1 and obtains L_1's $level_1$ routing table. L_0 can make use of that $level_1$ routing polygon by forwarding to L_1. L_0 adds those $level_1$ entries to its routing table, noting that, when such a route is used, the packet is encapsulated and forwarded directly to L_1. L_1 decapsulates and routes the packet upon reception.

Assume all $level_0$ nodes follow that procedure. Upon its conclusion every node in $level_0$ will have a route to its closest $level_1$ node, the ability to make use of that node's routes and can have no more than thirteen entries in its route table [6+1+6].

Apply the above procedure recursively. A proportion of $level_n$ nodes are randomly chosen to also be $level_{n+1}$, nearest $level_n$ nodes are found, $level_n$ nodes add $level_{n+1}$ routes to their routing table, and so on. If the constant of proportionality is significantly less than one, the population in each level declines exponentially from that of the previous level and the procedure rapidly terminates when some level would have fewer than two nodes. In practice, no $level_n$ node would have more than $6(n + 2) + 1$ entries in its table. As long as $P \gg 1$, this hierarchy produces a diameter that is proportional to $\log_P(N)$, for a network of N routers and $N \gg P$.

V. Spatialcast and Bootstrapping

The reason to create a geophysical overlay is to provide services that are difficult or impractical to provide in an existing network. An example is spatialcast in the Internet, the broadcast of a packet into a defined physical region. The forwarding rules for spatialcast are now defined for circular and spherical regions for 1-hop consistent networks. Bootstrapping is then briefly discussed as are some open issues.

V.1. Spatialcast

Assume that spatialcast packets specify a destination location D and a radius. The radius and D define a destination space R. If a spatialcast is initiated from outside R, it may be forwarded as a unicast packet. Eventually, either the spatialcast reaches D or a node forwarding it is unable to make further progress. At that point, either that node is inside R or outside it. We proceed now to consider the normal case, in which a node inside R receives the spatialcast, and deal with boundary conditions later.

Any node in R that receives a spatialcast copies, marks it and transmits copies to all its directly connected neighbors. Duplicate detection is used to prevent loops. This procedure is sufficient to reach any node in R via a path that starts within and remains within R.

The first boundary condition occurs when D is not co-located with a node. It can then arise that a node N lies outside R and receives an unmarked spatialcast, but is unable to make further progress toward D. Call $\{I\}$ the set of nodes located inside R, which may be empty. All immediate neighbors of N must be located outside R, otherwise N could make continued progress toward D. Call that set $\{O\}$. All nodes in $\{I\}$ are closer to D than any member of $\{O\}$. Thus, since any member of $\{I\}$ is accessible via a path from N, any node in $\{I\}$ must be accessible via a path that makes progress toward D from at least one of $\{O\}$. Require that N forward marked copies of the spatialcast to all its immediate connected neighbors and that any node outside R that receives a marked spatialcast forward it to all neighbors that make progress toward D. Thus, any node in $\{I\}$ must eventually receive a marked spatialcast.

The second boundary condition occurs when a node N inside R transmits a marked copy of the spatialcast to an immediate neighbor that lies outside R. Call that set of nodes $\{O\}$ and call $\{I\}$ the set of nodes in R, possibly empty, that are not accessible from inside R via a path that lies wholly within R. Note that each member of $\{I\}$ is closer to D than any member of $\{O\}$. Require that any member of $\{O\}$ that receives a marked copy forwards it to any neighbor that makes progress toward D. Thus, all paths from R that emerge to a member of $\{O\}$ and that subsequently make progress toward D are traversed by a marked spatialcast.

V.2. Bootstrapping

Bootstrapping is now addressed: How might a host that wishes to join a GONet find its neighboring hosts? Two scalable approaches are outlined.

Assume that a host H is trying to join the geographic network in its area. H can use local broadcast to find any other geophysical host on its local network. If such a host exists, that host may proxy for H. H can ask the proxy to issue a spatialcast query on its behalf. The query response is the set of neighbors that surround a location within some radius r. This allows H to run the greedy edge algorithm. If the resulting polygon is closed, H can construct its routes and proceed to announce its presence over the network so that its neighbors can adjust their routes if needed. Otherwise, H can issue another proxy spatialcast to find hosts located in the unclosed region of its routing polygon. If H is not near the edge of the network, such hosts will be found.

The second bootstrap method makes use of a distributed database. A variation of the DNS database, hierarchically distributed by geophysical region, would provide an enquiring host with the set of neighbors that

surround a location within some radius r. In summary, reasonable and scalable mechanisms can be created that would allow a host to determine its set of neighbors and create its routing table.
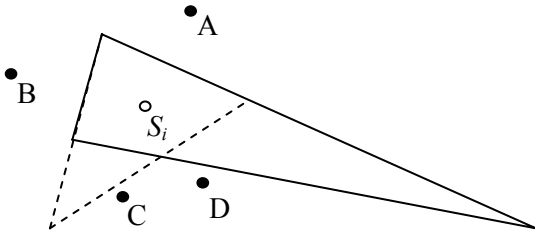

Figure 6 Corridor Problem

V.3. Open Issues

Greedy Edge generates small routing tables, but for some nodes S_i it can generate routing polygons that have far distant vertices. Determining that such a polygon contains no nodes other than S_i, and so is closed, can then require extensive searching. This 'corridor problem' is illustrated in Figure 6, where nodes A, B and C define the polygon. Issuing a spatialcast for large radii could cause a message implosion problem for the issuer and waste network resources. This search problem can be substantially mitigated by the use of small spatialcasts that tile the unclosed region of a polygon. As indicated by the figure, it may also be possible to sidestep the corridor problem by building a routing polygon from another set of neighbor nodes without increasing the number of polygon edges. The polygon determined by A, B and D results in a much closer farthest vertex. A heuristic to detect and fix the corridor problem is certainly possible, but has not been studied.

Because it is a broadcast service, unconstrained spatialcast presents a serious risk for a denial of service attack. The mechanisms needed to prevent such attacks have not been studied. It is presumed that network wide limits imposed by routers, such as radii limits and per-host issue rate limits would go a long way to solving the problem. Where unconstrained spatialcast is needed, such in emergency services applications, authentication can be used.

The limit in routing table size achieved by the greedy edge algorithm in three dimensions was not determined by the authors. Although simulation indicates a small typical bound, without a proven bound the possibility of large routing tables exists.

When greedy edge is applied recursively to reduce hop counts, each successive level of the hierarchy spans a greater area. At the highest network level it appears that a spatialcast of global scope is needed to determine where other nodes of the same level are located. A

straightforward way to eliminate this scaling problem is to extend spatialcast so that it forwards only to nodes associated with a specified level.

VI. SIMULATION RESULTS

The GONet protocols were validated using two independent simulations, one written in ns-2 (see www.isi.edu/nsnam/ns), and the other a custom Perl simulation. Both simulations confirmed, for a variety of sizes, the following results. Because every node in a GONet is a potential router, large ns-2 simulations were prohibitive both in size and performance; as a result, the following data are from the Perl simulation, though data for the smaller data sets yielded equivalent results under both simulators.

The Perl simulations were run on a set of five FreeBSD PCs with dual 1 Ghz Pentium-III processors, and the larger simulations took several days per run. Each run first computed the routing table for each node, then computed the hop count between each pair of nodes by emulating packet traversal through the network. Both two- and three-dimensional GONets were validated and measured; the results here focus on the 2-D data.

Each run tested 100-10,000 nodes placed in a uniform pseudo-random distribution on a square grid. No two nodes occupied the same grid position, as this simulation focused on the geographic routing, rather than the ID resolution mechanism. Each data point represents the average of 10 runs, with different random seeds (thus different node placements).

VI.1. Results

The simulations measured both the number of hops between each pair of nodes, as well as the size of the routing table of each node. For a flat GONet, the typical hop count increases as sqrt N. For a hierarchical GONet, the hop count increases logarithmically.

Figure 7 shows the cumulative distributions in a flat network of hop count vs. percentage of nodes reached, for graphs (left to right) of 100, 200, 300, 400, 500, 1,000, 2,000, 3,000, 4,000, 5,000, and 10,000 nodes. The graph suggests O(sqrt N) reachability; 85% of the nodes in the 100-node graph are reachable in 10 hops, 85% of the nodes in the 1,000-node graph are reachable in 30 hops, and 85% of the nodes in the 10,000 node graph are reachable in 100 hops. Further, the graphs of 1,000-5,000 are identical to the graphs of 100-500, shifted to the right, representing approximately 3x longer hop count (sqrt 10). The standard deviations are shown on this graph, but omitted from further graphs for clarity, and because, as in this graph, they are sufficiently small as to be negligible.
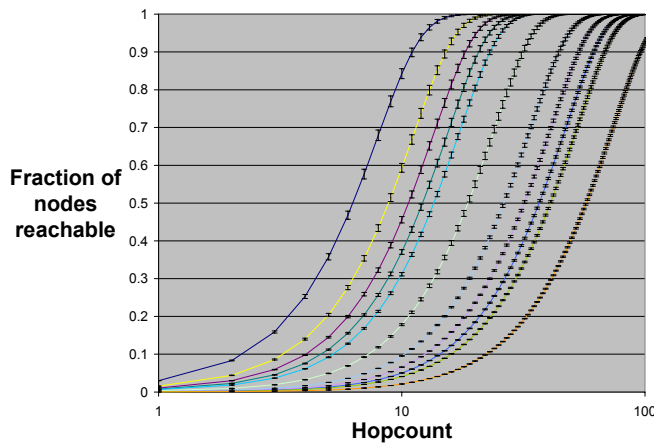
Figure 7 2-D GONet reachability vs hop count

By using GONet recursively (a GONet on a GONet), a routing hierarchy by distance is defined. The hop count can then be substantially reduced. Figure 8 shows the same sets of runs as Figure 7, but with recursive application of greedy edge routing, with one router in seven at $level_n$ chosen randomly to participate as a $level_{n+1}$ router. Thus, as network size increases, so does the number of levels. Note that in this graph, as the number of nodes increases, the graph becomes steeper. The graphs of 1,000-5,000 are clustered more closely than those of 100-500.
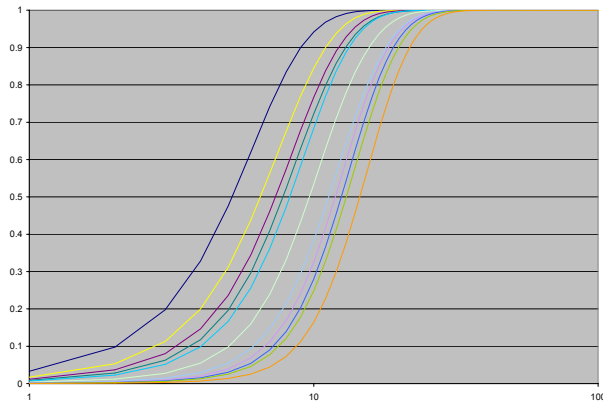

Figure 8 2-D, hierarchical GONet

To compare these results more precisely, consider the 80th percentile of the plots in both graphs, charted on a log-log scale (Figure 9). The top plot is the 80th percentile of the plots in the flat, 2-D GONet (Figure 7), shown with a curve predicting sqrt N growth. The bottom plot is the 80th percentile of the plots in the hierarchical, 2-D GONet (Figure 9), shown with a curve predicting log N growth. Although the curves do not indicate variance [we're working on computing the variance between two PDFs], the manual estimates show very good fits.
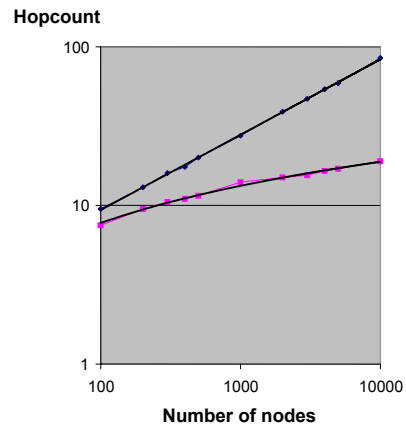

Figure 9 Analysis of hopcount growth

Both the ns-2 and Perl simulations verified that the size of the routing table is fixed in the flat GONet cases. As was shown previously, the upper bound is 6 entries; in practice, nodes had 3-4 entries, with a small percent having 5, and none showing the pathological 6 entry table; 2- and 1-entry tables occurred as boundary cases on the edge of the grid.

In the hierarchic case, the maximum bound on routing table length depends on the number of levels. For the 10,000 node networks no more than five levels were generated implying an absolute bound of 30. In simulation no router had more than 18 entries. Because progressively fewer and fewer nodes participate in higher levels, the mean routing table length for all nodes is expected to be much smaller and indeed was held to below 6.7 entries for all runs.

VII. RELATED WORK

Geographic unicast packet routing has been explored for use in both traditional and wireless networks. Use of geographic host addresses for non-wireless application was initially explored in the mid-1980s as a way of eliminating scaling difficulties associated with routing table size, routing protocol overheads, and to provide support for mobility in metropolitan-scale networks [1]. The advantage that geographic routing provides in robustness and security was also explored [4].

A network graph that is not geographically consistent can be made consistent through the addition of necessary edges. This was done in [1] for the ARPANET by flooding searches that constructed source route fix-ups for affected host routing tables, but that approach did not scale. The use of overlay technology eliminates the need to discover and maintain source routes.

A geographically consistent graph enables a spatial analogue to traditional multicast. However, unlike

multicast, spatialcast group membership is implicitly defined by current host location. This ability was suggested in [1] and has been explored as an application-layer protocol [5][6]. The principal difference between the spatialcast suggested here and the work on GeoCast by Imielinski and Navas is reliance on a consistent geographic network topology and unicast stack, rather than implementation in the application layer.

As GPS receivers become both commonplace and relatively inexpensive, it becomes practical to use geophysical location in traditional network settings and for routing in ad-hoc wireless and optical networks. The Spatial Working Group within the IETF has been developing application-layer protocol standards to allow geographic data querying [7]. Included in that effort are developing standards for the representation of geographic data and queries [8].

A very general, web-oriented application to allow capture and display of geographic information has been proposed by SRI that makes use of the DNS. Access to this database would be made via a top-level geo domain <http://www.sri.com/news/releases/10-23-00.html>.

Early versions of IPv6 reserved a portion of its 128-bit address space for geographic addresses. This reservation was later removed and now appears to have been replaced by the Provider-Independent Global Unicast Format [9][10]. This format interleaves latitude and longitude magnitude information into a network number suited to the longest-match routing used in the Internet. With the lower 64-bits of the IPv6 address, plus FP and SLA fields reserved, all geographic data must fit into a 44-bit field. This determines a resolution grid limit of approximately 10 meters per side. While that is sufficient for current GPS system limitations, it is extremely coarse compared to land title benchmarks. Another weakness is the absence of Z-axis information. Without the ability to discriminate in three dimensions, all floors in a building are addressed simultaneously. These limitations would place serious constraints on applications.

A principal advantage of wireless hosts is mobility. But mobility implies a continually changing network topology. Constant topological flux makes inefficient routing algorithms that depend upon the widespread exchange of distance vector or link state information. This has led to development of routing protocols that do not continually and globally circulate topology changes [11][12][13]. Scalability concern remains as both the size of the ad-hoc network and host mobility rate rises. Geographic addressing allows progress-based forwarding, which was suggested because routing then need rely only upon local information. Where the

constraint of local information causes a forwarding void, a perimeter forwarding method has been developed that still relies only upon local topology information [14].

Geographic information has been used to construct routing tables to minimize transmission power [15], and geophysical addressing has been applied in wide-area wireless networks to route while minimizing power usage [16]. A combination of modified Manhattan topology and Cartesian routing has also been recently suggested for application to optical networking [17], where routing based upon local geographic topology data reduces routing table size so as to allow forwarding in O(1) time.

As was mentioned earlier, a geophysically consistent graph can be based upon constructing Voronoi regions for each node in the network. Once determined, Delaunay triangulations define the edges and routing table for the nodes. Voronoi diagrams and Delaunay triangulations are a subject of study in Computational Geometry [2][3]. Voronoi diagrams have been suggested for creation of routing tables in wireless networks and to implement spatialcasting [18].

The amazing ability for people to form short chains of acquaintances that connect pairs of people who do not know one another was experimentally determined in a series of experiments carried out by Stanley Milgram's group in the 1960s. This verified what was known as the small-world phenomenon [19]. The use of graph theory to model that behavior has been the subject of recent studies. One drawback was that while it is possible to create large networks with small average node degree that allow short paths between any two hosts, the knowledge of which links to follow at each hop to obtain a short path required global knowledge.

Kleinberg determined that relatively short paths could still be obtained for large networks with a small node degree, even though only local knowledge is presumed at each node [20]. This was achieved by assuming a Manhattan directional metric and adding a small number of 'long-distance' links distributed randomly according to a distance-sensitive probability distribution. However, although routing decisions no longer need global information to generate short paths, global knowledge appears needed to generate the distribution function properly.

VIII. CONCLUSIONS

The ability to address hosts by their physical position has obvious application to military, emergency services and commercial organizations. As GPS and personal network communications devices become popular, the

need for position-sensitive communication will rise dramatically. However, Internet addressing and routing is organized by administrative domain rather than by geographic location, making it very difficult to develop applications that rely upon geophysical position or proximity. The broadcast of packets to hosts that lie within in a defined geographic region (*spatialcast*) is currently incompatible with Internet routing.

A methodology has been presented that assigns participating hosts two- or three-dimensional geographic addresses. A greedy graph-generating algorithm was developed that generates a network interconnection graph organized by physical host proximity, that produces a complete, consistent network and that for two-dimensional addresses requires no router to have more than six entries in its routing table. Completeness, consistency and routing table bounds were validated by extensive simulation for networks of up to 10,000 nodes. Hop counts in a flat geographic network were determined to be proportional to the sqrt of network size. Recursive reapplication of the graph-generating algorithm produced a routing hierarchy that was shown to produce typical hop counts log proportional to network size.

Implementing such a geographic network as an overlay enables position-sensitive applications and spatialcast without requiring any changes to underlying Internet protocols. Such an overlay makes use of IP protocol encapsulation (tunnels). If a geographic routing hierarchy is used, Internet connectivity is required only for the lowest hierarchy level. Unlike the Internet the routing tables at the core of this hierarchy would be very small.

IX. ACKNOWLEDGEMENTS

X. REFERENCES

[1] Finn, G. G., "Routing and Addressing Problems in Large Metropolitan-scale Internetworks", University of Southern California/Information Sciences Institute, ISI/RR-87-180, Mar. 1987.

[2] Preparata, F. P., Shamos, M. I. Computational Geometry: An Introduction, Springer-Verlag New York, Inc., 1985.

[3] de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O., Computational Geometry: Algorithms and Applications, Springer-Verlag Berlin, 1997.

[4] Finn, G. G., "Reducing the Vulnerability of Dynamic Computer Networks", University of Southern California/Information Sciences Institute, ISI/RR-88-201, Jun. 1988.

[5] Navas, J. C., Imielinski, T., "GeoCast - Geographic Addressing and Routing", Proc. of 3rd ACM/IEEE International Conf. on Mobile Computing and Networking, Sep. 1997.

[6] RFC 2009: GPS-Based Addressing and Routing Imielinski, T., Navas, J., Nov. 1996

[7] Loughney, J., Costa-Requena, J., "Basic SLoP Architecture Proposal", Internet Draft, Jul. 2000.

[8] Mahy, R., "A Simple Text Format for the Spatial Location Protocol", (SLoP),Internet Draft, Jan. 2001.

[9] An IPv6 Provider-Independent Global Unicast Address Format T. Hain, Internet Draft, Mar. 2002.

[10] Application and Use of the Provider Independent Global Unicast Address Format, T. Hain, Internet Draft, Mar. 2002.

[11] Johnson, D. B., Maltz, D. A., "Dynamic Source Routing in Ad-hoc Wireless Networks", In Ch. 5 of Mobile Computing, T. Imielinski, H. Korth, Eds. Kluwer Academic Publishers, 1996.

[12] Perkins, C. E., Royer, E. M., Das, S. R., "Ad hoc On-Demand Distance Vector (AODV) Routing" Internet Draft, Nov. 2000.

[13] Haas, Z. J., "A New Routing Protocol for the Reconfigurable Wireless Networks", ICUPC '97, San Diego, CA, Oct. 1997.

[14] Karp, B., Kung, H. T., "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", Proc. of the 6th ACM International Conf. on Mobile Computing and Networking, Aug. 2000.

[15] Stojmenovic, I., Xu Lin, "Power Aware Localized Routing in Wireless Networks", IEEE International Parallel and Distributed Processing Symposium, Cancun, Mexico, May 1-5, 2000, pp. 371-376.

[16] Kalkworf, R. L., Baran, P., Flammer III, G. H., "Method and system for routing packets in a packet communication network", US Patent EP0455959, Metricom Corp., Nov. 1991.

[17] Hughes, L., Banyasad, O., Hughes, E., "Wide Area Cartesian Routing", Computer Networks, Vol. 34, No. 3, Sep. 2000.

[18] Stojmenovic, I., "Voronoi Diagram and Convex Hull Based Geocasting and Routing in Wireless Networks", TR-99-11, School of Information Technology & Engineering, University of Ottawa, Canada, 1999.

[19] Milgram, S., "The Small World Problem", Psychology Today, 1, 61, 1967.

[20] Kleinberg, J., "The Small-World Phenomenon: An Algorithmic Perspective", Proceedings of 23nd Annual ACM Symposium on Theory of Computing, May 2000.

[21] Touch, J., "Dynamic Internet Overlay Deployment and Management Using the X-Bone", Computer Networks, Jul. 2001, pp. 117-135.