

A Locality Preserving Routing Overlay using Geographic Coordinates

Shah Asaduzzaman and Gregor v. Bochmann
School of Information Technology and Engineering
University of Ottawa, Ottawa, ON, Canada K1N 6N5
{asad,bochmann}@site.uottawa.ca

Abstract—This paper presents a design for a general locality preserving routing overlay network based on geographic coordinates of nodes. For efficient use of network resources, it is important to follow some locality principles while routing bulk multimedia content. The proposed overlay network creates the interconnection among large number of highly dynamic nodes in a completely decentralized manner, based on adaptive hierarchical partitioning of the geographical space. A major benefit of using coarse grain geographic coordinates is that they are available at off-the-shelf databases and thus do not incur additional measurement overhead. The performance of the proposed geography based routing overlay, in terms of two locality properties – stretch and route-convergence, is evaluated by simulation studies. Real topology data of 65 ISPs with presence in 534 cities across the world, collected by the RocketFuel project, is used as the basis of the simulated network. The performance results are compared with corresponding results obtained from a simulated Pastry overlay constructed on the same underlying network. The performance results show that proposed overlay has significantly better stretch and route-convergence characteristics compared to a regular Pastry network. The locality properties of the proposed routing overlay are almost equivalent to those of an optimal Pastry network where all nodes use the most proximal neighbor for each routing table row. Thus the proposed geographical routing network provides good locality properties without the overhead of proximity neighbor selection. Moreover, the proposed scheme facilitates certain geographical search applications that are difficult to realize in a Pastry overlay.

I. INTRODUCTION

In the peer-to-peer literature, originated from the study of decentralized file-sharing systems, *peers*, or Internet-connected end-hosts, communicate through a self-organized network of acquaintances, called *overlay network*. These overlay networks are nowadays envisioned as a single self-organized and decentralized substrate to be used by many different large-scale networked applications. Progress have been made in this direction and several well-understood features of a generalize peer-to-peer substrate have been defined [1], [2].

The evolution of peer-to-peer research has historically depended on the content sharing applications. Two fundamental functional requirements of these applications – search and transport have guided the design of general purpose overlays. Initially, the problem of search or lookup dominated the research while the transport related problems gradually emerged. Researchers then have attempted to accommodate transport related functionalities in generalized versions of the overlays that were primarily created for lookup [3]. The class of overlay networks designed for

looking up data content using data-keys are often called *distributed hash tables* (DHT) for the hash-table-like get and put interface they provide. The DHTs perform the lookup operation in a decentralized way by routing the lookup message through the structured acquaintance network. Such key-based routing has been proposed as the standard service interface that can be used to derive the necessary functions for different applications [1].

A common design decision taken by the designers of the overlay structures, primarily designed for the lookup service, is randomly assigned flat numeric identifiers for the hosts, while the overlay structures are defined in terms of the numeric properties of the identifiers. The arguments placed in favor of such randomness include load balancing, placing of replicas at uncorrelated hosts and anonymizing the source of a content. While accepting that all these features are necessary for content sharing, researchers have argued that randomized placement of hosts in the overlay structure is not the only way, nor the best way, of achieving them [4]. More importantly, message routing in these random-identifier based overlays ignores the conventional wisdom of considering locality for routing in order to obtain efficient usage of network resources.

Ever since the emergence of the peer-to-peer content sharing application, there have been growing complaints and consequent policing from the Internet service providers (ISP) on the traffic generated by peer-to-peer applications. Though part of the reason of this overwhelming traffic is the sheer volume of the contents, the randomized message routing topologies of most of the peer-to-peer overlays also shares part of the blame. In response to the complaints from the ISPs, there have been several proposals for introducing locality awareness in the overlay structures [5], [6], [7], [8], [9], [10] such that the messages routing follow some locality principles [11]. In most of the cases, localities are defined based on explicit measurements of some application level metric such as latency. This class of overlays, denoted as network aware overlays suffer from the large background overhead of the measurement.

In this paper, we argue that the geographic location of the end-hosts, usually available off-the-shelf at the granularity of ISP's points of presence (POP), can be used as the basis of a locality-based routing overlay. The argument is founded on the observation that the Internet infrastructure has significant geographic clustering and hierarchical organization [12], [13]. A further motivation of using geographic

coordinates is that the major fraction of the acquaintances in online social networking communities are dictated by geographic proximity [14]. It is visible that the data exchange between socially affiliated peers dominates the cyber-traffic today, from sending messages, emails and blogs to sharing videos, photos and music. If geography is a major factor behind social clustering, then it is important to optimize the network structure for geographically localized communications.

The main contributions of this paper is an overlay structure capable of locality-based routing constructed using geographic coordinates of the peers. The overlay can be used as a routing substrate for different applications where locality-based routing is useful, such as multicasting, content delivery networks, operating and data-gathering from sensors and in-network processing of data streams generated by sensors. The properties of the proposed overlay structure are evaluated using a simulated network built based on the POP-level topologies of 65 ISPs traced by the RocketFuel project, covering 534 cities across the world, mostly in United States, Europe and Australia [15], [16]. The study provides evidence that in the proposed overlay structure, built solely based on geographic coordinates without knowing the underlying network metrics, the routes of the messages maintain very good locality properties in terms of the underlying network.

The rest of the paper is organized as follows. Section II explains the locality properties that are important for different applications. Section III introduces the proposed overlay structure, and explains how messages are routed and how the structure is maintained in presence of peer arrival and departure. Section IV presents the simulation study of the overlay that evaluates its locality properties.

II. THE LOCALITY PRINCIPLES

When large volumes of traffic are transmitted between nodes of a network, it is important to use the network resources efficiently. From the experience of IP routing, it is understood that if some simple principles are followed in local decisions, the desired global properties emerge.

One such well-known principle is the principle of locality, which requires that the transport path between two endpoints of the same local region should remain within the region [11]. This discourages the traffic to take arbitrary detours causing unnecessary burden on the global network.

In the large scale content sharing application, the same content is transported to a large number of end-points, either at the same time or asynchronously over an extended period of time. This requires an overlay network that support efficient construction of multicast trees. Other applications such as in-network processing of sensor data-streams also require placement of the data-processing operators based on the multicast trees.

A locality property that results in efficient resource usage in multicasting is the path-convergence property, which states that paths from a single source to multiple destinations in one locality should have significant portion of the path shared. The smaller the area of the locality, the larger should

be the common segment. Intuitively, this can be attained, if the localities are hierarchically divided, and the traffic follows a direction towards destination, gradually resolving the destination at a deeper level of the hierarchy. Such directional routing with hierarchical resolution will be explained in further details in Section III.

III. STRUCTURE OF A GEOGRAPHY-BASED OVERLAY

In this section, we introduce an overlay interconnection structure named GeoP2P, based on hierarchical partitioning of the geographic space, where traffic is routed towards the geographic location of the destination, successively resolving the destination at a deeper level of the hierarchy.

A. Structure

The universe (earth surface) is hierarchically divided into zones, sub-zones, sub-sub-zones and so on. Each peer is assumed to know its own geographic coordinate (latitude, longitude). A zone is divided into non-overlapping sub-zones and the higher-level zone completely covers all the areas of its sub-zones. The shape of the zones need to be amenable to concise memory representation and also to easy computation of whether a point belongs to a zone or not. We used axis-parallel rectangles as the shape of the zones. At the leaf level of the hierarchy are the zones that are not divided any further (denoted *leaf zone*). Each individual overlay node or *peer* belongs to a leaf zone at its deepest level, to successively larger zones at higher levels, and to the zone covering the universe at the top level. Figure 1 illustrates an example division of the universe and the corresponding tree representation is shown in Figure 2.

A *routing table* in each peer stores the overlay neighbors of the peer, i.e. the list of the peers that this peer knows and can directly communicate with. A message is routed towards a destination by successively resolving the zones at finer grain. To be able to do so, a peer needs to know at least one peer from each of the sibling zones at every level of the hierarchy. This peer is denoted as *contact peer* for the corresponding sibling zone. At the deepest level, the peer knows all other peers within its own leaf zone. The routing table may be organized in rows, each row storing the pointers to the siblings at a given level. For each pointer, the IP address of the target peer and the boundary definition of the corresponding sibling zone is stored. Also, the boundary definition of the self-zone at every level is stored at the corresponding row. The overlay neighborhood of a peer is illustrated in Figure 2. Clearly, the state information maintained in the routing table is proportional to the depth of the hierarchy, which is roughly logarithmic to the total number of peers in the system.

In fact, the interconnection structure of the proposed overlay follows a small-world interconnection topology very similar to the one used in many DHTs with hierarchical structures [17], such as Pastry [18] and Kademlia [19]. The DHTs use Plaxton's prefix matching algorithm for routing [20], where a message targeted to a numeric key is routed through peers with numeric identifiers having successively longer prefix-matches with the key. The numeric identifier space used in the hierarchical DHTs can be thought

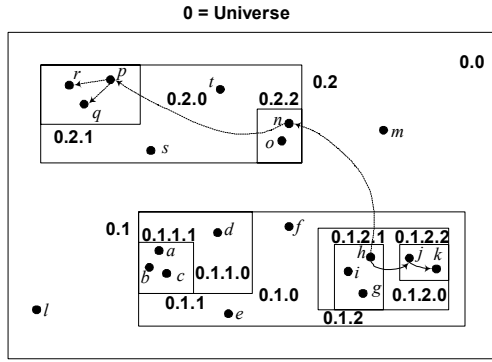


Fig. 1. Zoning of geographic clusters. Routing paths show locality ($h \rightarrow k$) and path convergence ($h \rightarrow (q, r)$)

being hierarchically divided into zones and sub-zones, each zone containing all the peers with identifiers of a certain prefix [17]. In that sense, the routing table of a peer in those DHTs also maintains pointers to at least one peer in all other sibling zones at each level of the hierarchy. The size of the routing table is similarly logarithmic to the total number of peers in both cases. The difference between the hierarchical DHTs and the proposed geographic overlay is that we have zoned the geographic space instead of the identifier space, and also our zoning hierarchy is dynamic instead of being pre-configured.

B. Adapting with the peer dynamics or churn

The beauty of the overlay structure lies in its flexibility to grow and retract with the membership dynamics (peer join and departure, or *churn*, and its ability to manage this in a completely decentralized manner.

1) *Growth*: When the number of peers in a leaf zone grows beyond a threshold θ_H , new sub-zones are created by dividing the zone according to geographical clusters of peers. As the new sub-zones are created based on geographic clustering, the area of the previous leaf zone that does not belong to any of the clusters, is also considered a sub-zone and is denoted as remainder zone. A remainder zone is always a leaf zone. The non-remainder zones are denoted as rectangular zones.

After division of a level d zone into k sub-zones of level $d+1$, each peer needs to update its routing table entries for level d and $d+1$. The new level- $(d+1)$ entries will be a subset of the previous level- d entries, pointing to only those peers that are located within the same level- $(d+1)$ zone. $k-1$ entries from the remainder of the previous level- d entries will fill the level d of the new routing table. Because the boundaries of the other $k-1$ level- $(d+1)$ zones are known, the peer can randomly choose one of the previously known peers for each of these zones. In addition, zone boundaries of those $k-1$ zones are stored in these entries, instead of the point coordinates previously stored. The remaining entries in the routing table can be discarded.

Note that all peers in the same leaf zone are neighbors to each other in the overlay. Also, a peer knows the areas corresponding to its sibling zones at each level of the hierarchy. So any peer of a leaf zone is able to perform the

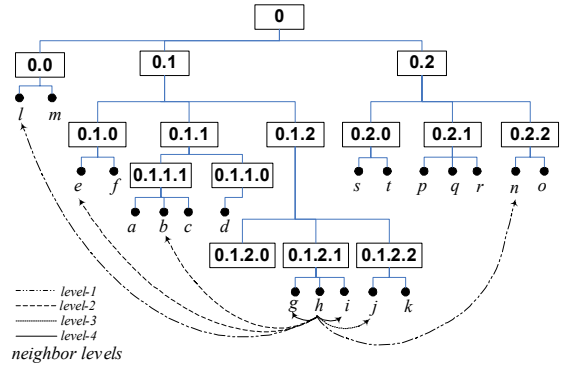


Fig. 2. Zoning hierarchy and overlay neighborhood of peer h

partitioning. To avoid concurrency problem, the peer that detects the need for partitioning initiates a one-round leader election algorithm. The leader then performs the partition operation and inform others peers of the new boundaries and identifiers.

From the procedure discussed above, it is obvious that the routing tables of all the peers can be updated very easily in only one round of message exchange, transmitting at most only $\theta_H - 1$ messages. The content of the messages is also very small; only the boundaries of the k newly formed zones need to be communicated. The routing table of only those peers that belong to the divided zone need to be updated, the maximum number of which is θ_H . Peers outside the zone are not affected. The computation done at each peer is also very simple and perturbs only the last two rows of the routing table. To minimize alteration of the routing tables, zone boundaries are not modified once zones are created. The only permitted ways to adjust the number of peers in a zone are dividing into sub-zones or merging with sibling zones.

2) *Retraction*: Similarly, when it is discovered that the number of peers in a leaf zone is below a threshold θ_L , the leaf zone initiates a merge with one of its siblings. Because the peers in the merging zone know about the boundary and at least one peer of each of the sibling zones, those zones become natural candidate for being merging partner. The remainder-leaf always represents as a suitable merging partner for its sibling leaf zones.

The peer that initiates the merger (denoted as *initiator*) knows at least one peer in the partner zone (denoted as *partner peer*). The partner peer knows all other peers in the partner zone. The initiator sends a *merger request* to the partner peer. The request contains the boundary of the merging zone and address and coordinates of all peers in that zone. On receiving the request, the partner peer realizes that it needs to extend the boundary of its own leaf zone and include the peers given in the message as neighbors. Say, both the merging and the partner zones are at level d of the hierarchy. So, the partner peer also needs to update its level $d-1$ of its routing table, by removing the entry corresponding to the merging sibling.

Besides updating its own routing table, the partner peer

also forwards the merger request to all other peers in its zone, so that all of them perform a similar update in their routing tables. The initiator, on the other hand, needs to send a *merger update* message to each of the other siblings (except the merging partner) so that peers in them remove the merging zone from level $d - 1$ of their routing tables. The initiator knows at least one peer in each sibling zone, so it can transmit the message to the known peer, which in turn can broadcast the message to all peers in its zone. Lastly, the partner peer needs to respond to the initiator peer by sending the address and coordinates of all other peers of the partner zone, so that all other peers in the merging zone can add them to their level d entries in their routing tables.

C. Routing

The overlay is able to route messages towards a geographic location. The target may be all or any peer(s) in the specified area, or the nearest or a nearby peer of a specified point. Messages may also be routed to a particular peer specified by its geographic location and an additional identifier to resolve among the peers having same geographic location.

Algorithm 1 RouteToPeer(*msg*, *location*, *uid*, *level*)

```

1: if peer.uid = uid then
2:   Deliver msg to peer
3:   return
4: end if
5: found  $\leftarrow$  false
6: if level  $\leq$  deepest level d then
7:   for Each entry e in row d of the routing table do
8:     if e.uid = uid then
9:       Send new RouteToAllPeers(msg, location,
10:      uid, d + 1) to e.IPAddress
11:      found  $\leftarrow$  true
12:      break
13:     end if
14:   end for
15: if found then
16:   return
17: end if
18: for r = d - 1 down to level do
19:   for each entry e in row r of the routing table, except
20:   for the one denoting self zone do
21:     if e.zone_boundary contains location then
22:       Send new RouteToAllPeers(msg, location,
23:      uid, d + 1) to e.IPAddress
24:       found  $\leftarrow$  true
25:       break
26:     end if
27:   end for
28: if found then
29:   break
30: end if
31: end for

```

To forward a message targeted to a node with unique id

uid and geographic coordinate *location*, a peer uses the *RouteToPeer* method defined in Algorithm 1. The peer first verifies if the target location is included in its leaf level self-zone (Lines 7-13). If so, the target peer is known (i.e. it is in the routing table), and the message is directly forwarded to the target. Otherwise, the routing table is scanned at successively higher level until a sibling zone is found with an area that includes the target location. The message is then forwarded to the contact peer for the matching sibling zone (Lines 18-29). To remember the levels of hierarchy already resolved, the *level* parameter is used, which is set to 1 at the peer that initiates the routing. So the loop in Lines 18-29 scans the rows for levels deeper than *level*.

Routing a message to all the peers in a specified area (not necessarily rectangular) can be performed by an algorithm very similar to Algorithm 1. The algorithm takes an *area* as the second parameter instead of a *location*. The condition in Line 8 checks for intersection with the target area instead of inclusion the target location. The loop in Lines 18-29 continues for all levels up to *level*, instead of breaking as soon as a match is found. That is, the message is forwarded to contact peers in each of the sibling zones at all levels of the routing table, whose zone-area intersects with the target area. Similarly, for the loop in Lines 7-13, the message is routed to all peers within the leaf level self-zone that fall in the target area.

If the peer nearest to a specified point is sought, it can be done by first reaching the peer that is closest to the peer within the zone that holds the point, and then sending a query message towards a circular area with the target point at its center and the current peer at the perimeter, to figure out if any other peer closer to the target exists.

By construction, it is observable that the overlay routing adheres to both the locality and the path convergence principles outlined in Section II. An illustration of the routing paths in Figure 1 demonstrates both the properties. Further details on maintenance of the overlay structure, including comparison with other hierarchical overlays such as [21] can be found at [22].

IV. EVALUATION

In this section we evaluate the locality properties of the proposed geography-based overlay through simulation.

A. Simulation Setup

As the basis of the simulation, we used the actual topology of a section of the Internet. Our goal is to evaluate whether the overlay structure constructed based on geographic coordinates maintain good routing locality in the underlying network. For our purpose, we used the POP level topology data collected by the RocketFuel project [16], [15]. The data set includes POP level topology of 65 different ISPs, of which 22 are tier-1 ISP, 32 are tier-2 and 11 are tier-1. They have presence at 534 different cities across the world, mostly in US, Europe and Australia. Multiple ISPs have presence at the same city, and in total we have 2098 POPs for all the ISPs. The links in the intra-ISP topology are annotated with latencies estimated from direct geographic

distance. Inter-ISP links are peering of the POPs of two ISPs at the same city, annotated with a 0 latency. The end-to-end route in the underlying physical network is assumed to be the latency-weighted shortest path.

The POPs are annotated with city names. We extracted the (longitude, latitude) coordinates of the cities using the Google Maps API [23] at city level granularity. We assumed a single hypothetical router for each POP, located at the center of the city. Peers or end-hosts are synthetically generated at all cities. The geographic location of the peers are generated randomly with a special distribution such that the probability of a peer being x km away from the city center is exponentially decreasing with x , and 95% of the peers are within a 20km radius from the city center. We assumed that the POP routers from all ISPs have the same number of subscribers; thus a city with a higher number of ISPs has a higher peer population. Although this may not be the right estimate of peer population, at a global level this is a good approximation of geographic distribution of peers. End-hosts are assumed to be directly connected to the POP router of its ISP at the city center, with path latency estimated from geographic distance. The overlays were constructed with 10000 peers.

With these peers, we constructed a geographic location based overlay according to the protocols presented in Section III (we will refer to it as GeoP2P). For comparison, we created a Pastry [18] overlay with the same set of peers and underlying topology. Each peer is randomly assigned a unique 32-bit numeric id, and the peer-ids are uniformly spread across the id space. At each hop of routing, 4-bits of prefix are matched with the key. Each 4-bit is called a *digit*. The row r of the routing table, r starting from 0, stores addresses of peers that match r digits of prefix with the current peer's id. Column c of the row r stores address of a peer that has $r+1$ -th digit equal to c . Thus each row of the routing table has 16 columns. It is observable that for the lower values of r , for each entry, there is a large number of peers to choose from. In standard Pastry, an arbitrary peer is chosen as long as it is live. There is also a proximity measure implemented in Pastry, which biases the choice for each entry in the routing table towards the closer peer among the possible choices. The proximity metric is usually an approximate measure of network latency. For the GeoP2P overlay, the θ_H and θ_L parameters are set to 32 and 16 respectively.

To evaluate how proximity neighbor selection can enhance the locality properties, we created the proximity enhanced versions of both pastry and GeoP2P (referred as Pastry-P and GeoP2P-P). In these cases each peer chooses the most proximal peer among several possible options for each entry in the routing table.

B. Results and Discussion

The first measure of locality is how closely the overlay route between two endpoints matches the direct end-to-end route. For this, the *stretch* of the overlay paths is measured as the ratio of the latency of the overlay route to the direct end-to-end latency. Figure 3 shows the probability distri-

bution of stretch values for the overlay paths between 5000 random pairs of nodes. Without any proximity enhancement, the GeoP2P topology has much lower stretch compared to the Pastry topology, because each overlay hop in Pastry may travel arbitrarily long distance. GeoP2P has very low stretch, even with random choice of routing neighbors, because the zoning structure itself enforces geographic proximity. The best possible stretch is attained by both GeoP2P and Pastry when the choice of routing neighbors is based on latency-based proximity. Note that this is an idealized case, and the real networks may not converge to these most optimal choices of neighbors. Moreover, there is measurement overhead to choose neighbors based on proximity, and the proximity neighbor selection potentially leads to load imbalance by increasing the indegree of some particular peers.

For multicasting or stream processing applications, a very important property of the routing overlay is route-convergence, i.e., when a message is routed from a single source to two different destinations, what fraction of the two paths are common. We measured the route convergence in terms of a convergence factor. Say the route between the source and both the destinations has d common underlay hops. The route to the first destination has additional d_1 underlay hops and the route to the second destination has additional d_2 underlay hops; then the convergence factor is defined as $(\frac{d}{d+d_1} + \frac{d}{d+d_2})/2$, which equals 1 when both the paths converge wholly (in case the two destinations are the same), and 0 when the two paths have no common segment. A high convergence factor is good because it means that less underlying network capacity is used for the transmission to both destinations. One expects that the convergence factor has a strong negative correlation with the distance between the pair of two destinations.

Figure 4 shows the route convergence factor for destination pairs at different distances. In case of GeoP2P, the convergence factor is pretty high, and has a strong inverse correlation with the distance between the destination peers. The proximity enhancement has little effect on route convergence. In case of Pastry, in fact, there is very little route convergence when a message is routed from a single source to multiple destinations (not shown in the figure). However, if the message is routed on the reverse path, i.e. when messages from two sources are routed towards a single destination, several terminal hops of the paths become common. When proximity-based neighbor selection is active in Pastry, it is more probable that the route from two nearby source towards a single destination converge into a single peer within initial few hops. Also, in Pastry, the later hops are exponentially longer than the earlier hops. Figure 4 shows the reverse path convergence in case of Pastry. Without proximity enhancement, the convergence is very low. Convergence increases with proximity enhancement, but remains somewhat less correlated to the distance between source peers.

In summary, the analysis shows that the GeoP2P overlay maintains strong locality properties in its routing, regardless of any additional proximity measure for selection of the

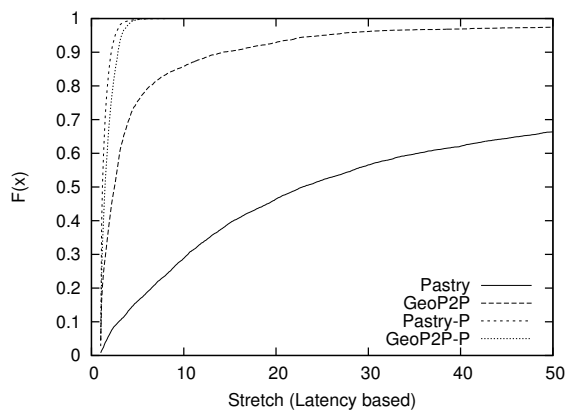


Fig. 3. Probability distribution of stretch (latency based)

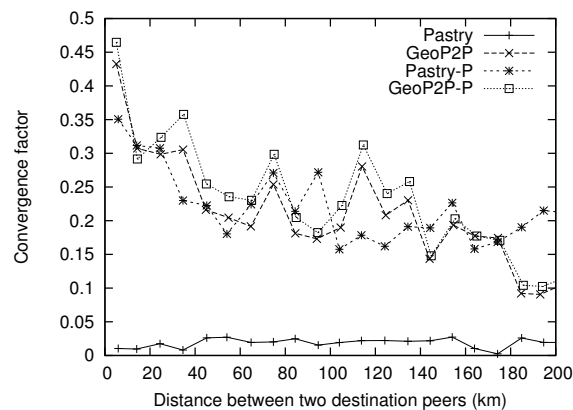


Fig. 4. Route convergence with distance between two destination peers

neighbors in the routing tables.

V. CONCLUSION

We have presented the design of a general distance-sensitive routing overlay based on the geographical distance of peer nodes. The routing structure is based on adaptive hierarchical partitioning of the geographical network space. Messages can be routed to a peer at a given geographical position; in this case this position plays the role of peer identifier. But a message may also be addressed to the set of all peers within a two-dimensional geographical region which is given as message destination.

We consider as main advantages of the geographic routing overlay that (a) it shows good values for stretch and convergence even when the peers in the routing table are selected randomly, and (b) it not only supports key-based searching (using the geographic position as search key), but also geographical range queries by allowing a request message to be routed to all peers that reside within a geographical region defined by the query.

We have presented here the design and evaluation of a specific overlay structure based on geographic coordinates of the peers, to demonstrate the usefulness of geographic location as a proximity measure. Alternative structures are possible to support geography based overlay routing. A comparative study of alternative designs would reveal further insights into the structures.

REFERENCES

- [1] F. Dabek, B. Y. Zhao, P. Druschel, J. Kubiatowicz, and I. Stoica, "Towards a common api for structured peer-to-peer overlays," in *2nd IPTPS*, 2003, pp. 33–44.
- [2] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, "OpenDHT: a public DHT service and its uses," *ACM SIGCOMM Computer Communications Review*, vol. 35, no. 4, pp. 73–84, 2005.
- [3] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "Scribe: A Large-Scale and Decentralized Application-Level Multicast Infrastructure," *IEEE JSAC*, vol. 20, no. 8, pp. 1489–99, 2002.
- [4] S. Zhou, G. R. Ganger, and P. Steenkiste, "Balancing Locality and Randomness in DHTs," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-03-203, Nov. 2003.
- [5] X. Y. Zhang, Q. Zhang, Z. Zhang, G. Song, and W. Zhu, "A Construction of Locality-Aware Overlay Network:mOverlay and Its Performance," *IEEE JSAC*, vol. 22, no. 1, pp. 18–28, 2004.
- [6] H. Le, D. Hong, and A. Simmonds, "A Self-Organising Model for Topology-Aware Overlay Formation," in *IEEE ICC*, vol. 3, May 2005, pp. 1566–1571.
- [7] N. Ball and P. Pietzuch, "Distributed Content Delivery using Load-Aware Network Coordinates," in *Proceedings of the 3rd International Workshop on Real Overlays & Distributed System (ROADS'08)*, Madrid, Spain, 12/2008 2008.
- [8] P. Pietzuch, J. Ledlie, and M. M. Seltzer, "Network-Aware Overlays with Network Coordinates," in *26th IEEE ICDCS Workshops*, Jul. 2006.
- [9] Z. Xu, C. Tang, and Z. Zhang, "Building Topology-Aware Overlays using Global Soft-State," in *23rd IEEE ICDCS*, May 2003, pp. 500–508.
- [10] M. Waldvogel and R. Rinaldi, "Efficient Topology Aware Overlay Network," *ACM SIGCOMM Computer Communications Review*, vol. 33, no. 1, pp. 101–106, 2003.
- [11] P. J. Denning, "The Locality Principle," *Communications of the ACM*, vol. 48, no. 7, pp. 19–24, 2005.
- [12] M. T. Gastner and M. E. J. Newman, "The Spatial Structure of Networks," *The Eur. Phys. Journal B*, vol. 49, pp. 247–252, 2006.
- [13] S. M. Greenstein, "The Economic Geography of Internet Infrastructure in the United States," in *Handbook of Telecommunications Economics*. Elsevier, 2005, vol. 2, ch. 8.
- [14] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins, "Geographic Routing in Social Networks," *Proc. National Academy of Sciences*, vol. 102, no. 33, pp. 11 623–11 628, 2005.
- [15] N. Spring, R. Mahajan, and T. Anderson, "The Causes of Path Inflation," in *SIGCOMM '03*, 2003, pp. 113–124.
- [16] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with rocketfuel," *IEEE/ACM Trans. Networking*, vol. 12, no. 1, pp. 2–16, 2004.
- [17] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The Impact of DHT Routing Geometry on Resilience and Proximity," in *SIGCOMM '03*, 2003, pp. 381–394.
- [18] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *IFIP/ACM Intl. Conf. on Distributed Systems Platforms*, 2001, pp. 329–350.
- [19] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *1st International Workshop on Peer-to-Peer Systems (IPTPS)*, Mar. 2002, pp. 53–65.
- [20] C. Plaxton, R. Rajaraman, and A. Richa, "Accessing Nearby Copies of Replicated Objects in a Distributed Environment," in *9th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, 1997.
- [21] A. Kovacevic, N. Liebau, and R. Steinmetz, "Globase.KOM - A P2P Overlay for Fully Retrievable Location-based Search," in *IEEE International Conference on Peer-to-Peer Computing (P2P-2007)*, Sep. 2007, pp. 87–96.
- [22] S. Asaduzzaman and G. v Bochmann, "GeoP2P: An adaptive peer-to-peer overlay for efficient search and update of spatial information," *CoRR*, vol. abs/0903.3759, 2009.
- [23] "Google Maps API," <http://code.google.com/apis/maps/>.