On Restricting Overhead of Location Lookup in Mobile Wireless Networks

Jun Liu
Computer Science Department University of North Dakota
Grand Forks, ND 58202-9015
jliu@cs.und.edu

Abstract

As location-based routing becomes useful in some particular scenarios, locations of mobile hosts become as important as their identifiers in ad hoc routing in mobile wireless networks. Location lookup services are henceforth needed to aid location-based routing.

In this paper, we present a source routing based location lookup service that can be used as the location discovery component in a location-based routing protocol. The design goal of this service lies in two folds: having high success rates on location discovery while keeping low demands on network resources. The goal of having high success rates is achieved through encouraging mobile hosts to cooperate in serving location queries. The goal of keeping overhead low is achieved through making each location query be served by a small number of hosts.

Each mobile host is associated with a number of friend hosts and distributes replicas of its up-to-date location to them in order to enhance chances of answering queries about its location. Discovering or updating the location of a target host is served through the cooperation among a set of friends of the target host. The friendship among mobile hosts forms into an index structure used in our location lookup service. In order to ensure a small number of hosts participating in serving a location query, we construct the index structure into a complete binary search tree and distribute it across mobile hosts. DSR routing protocol has been modified and used in forwarding location lookup service related packets. The path of forwarding packets reflects underlying temporary connectivities among mobile hosts.

Evaluated by simulating mobile wireless networks, the performance of our location lookup service exhibits high success rates and low demands on network resources.

Jun Liu
Computer Science Department
University of North Dakota
Grand Forks, ND 58202
jliu@cs.und.edu

1 Introduction

Geographical proximity plays an important role in ad hoc routing in mobile wireless networks. If two mobile hosts are geographically close together, there is a high probability that they can establish a direct communication link. Location-based routing protocols make use of locations of mobile hosts to determine next-hop hosts in forwarding packets to destinations. In contrast, geographic proximity in wired networking is not necessarily useful. For example, in the Internet, a packet can be forwarded in one hop to go across the Atlantic ocean through the cross-the-ocean optical fiber cables, but it may take several hops to reach an area nearby if there is no direct connectivity.

The ability for a mobile host discovering its current location is assumed in location-based ad hoc routing. The example location-enabling technologies include GPS (the satellite-based positioning system), landmark-based positioning, and movement-based positioning in ad-hoc networking. In landmark-based positioning systems, like the APS [12], a number of landmark beacons (fixed in position or mobile) broadcast their accurate positions. A mobile host could determine its current location through geometric computations using the signal strengths received from different landmark beacons. In movement-based positioning systems, if a mobile host knows its initial position, then it could compute its position afterwards using its moving speed and directions.

In ad hoc routing protocols making no use of locations, routes are maintained mainly through two approaches: the table-driven approach and the source-initiated on-demand driven approach [17]. In the table-driven approach, each mobile host maintains routing table(s) to reflect changes in network topology and propagates its view of current network topology to other hosts. In the source-initiated on-demand driven approach, no routing table(s) is generally maintained at a mobile host. Instead, when a (source) mobile host wants to find out a route to reach a destination host, it initiates a route discovery process within a network. Once a route is discovered, the source host specifies the full path in each data packet sent to the destination host, and every intermediate host specified in the path forwards data packets to a next-hop host in sequence. When the destination host becomes inaccessible within the duration of a data transmission due to changes of network topology, the source host has to find a new route by initiating a new route discovery process. Hence, overhead on maintaining route information is inevitably introduced in either approach.

In location-based ad hoc routing, route discovery is not necessary if the location of a destination host is known to a source host, because packets can be geographically forwarded from a source host to a destination host. Thus, discovering the location of a destination host is the only pre-requisite for establishing a route to reach the destination host. In this paper, we propose a method of distribution of location information so to restrict the overhead of location lookup. The design goal of this method lies in two folds: to have high success rates on answering location queries and to have low demands on network resources. The two aspects of the design goal often conflict. On one hand, in order to have high rates of successful location look-ups, up-to-date location information is ideal to be propagated to as many hosts as possible, and doing so consumes considerable amount of network resources. On the other hand, in order to prevent from consuming too much network resources, the amount of location information propagated in the network needs to be limited, and doing so hurts success rates.

In order to simultaneously achieve both aspects of the design goal, location information needs to be propagated wisely. That is, location information still needs to be propagated, but the amount of propagation has to be restricted. We make each mobile host only propagate its up-to-date location to its friend hosts and make this friendship publicly understood by every host. A host determines its friends using host identifiers. Two hosts are friends if their identifiers share a common suffix. The longer common suffix of identifiers are shared between two hosts, the closer friends are them. The two closest friends are mutually called peers. Moreover, in order to restrict the amount of propagation, a host only propagate an update on its location to a small number of its friend hosts.

Relations of sharing common suffixes of host identifiers are naturally expressed into a complete binary search tree. In the search tree, leaf nodes represent host identifiers, and internal nodes represent sets of host identifiers sharing various-length common suffixes. Such a complete binary search tree is used as the index structure in our location lookup service.

Demands on network resources can be restricted by making use of this index structure. We make a host always propagate updates on its location to its peer host. When a host can not reach its peer host in one hop, intermediate hosts are selected to forward updates on its location. A host initiating a location update selects its closest friend in its current neighborhood as a next-hop host, *i.e.* it shares the longest suffix of identifiers with the next-hop host among all hosts which are one hop away from it. Each intermediate host along a forwarding path follows the same rule in picking up a next-hop host. Upon a host receiving a location update from its friend host, it can choose to memorize this location information received and be ready to answer queries about the location of this friend host. A location update moves closer toward the leaf level of the complete binary search tree each time it is further forwarded. Doing so restricts the number of times a packet of location update is forwarded, *i.e.* no more times than the height of the search tree. Likewise, a packet of location query is forwarded in a similar way, and thus, it is also forwarded no more times than the height of the search tree. Our location service puts low demands on network resources because each service packet is only forwarded a few times among a small number of hosts.

High success rates can also be attained by making use of this index structure, because a location query is always forwarded to a host which could answer this query more likely, *e.g.* a location query about a target host is more likely answered by the peer of the target host than other friends of the target host. A location query about a target host could be answered by any friend of the target host along a forwarding path. Meanwhile, attaining high success rates also closely relates to the underlying temporary topology in a network. The requirement that a forwarding only happens between a pair of hosts having a direct communication link between them makes a forwarding chain likely broken before a location query is answered or a location update is propagated to its final destination. In a network with a high density of hosts, forwarding chains are unlikely broken, whereas, in sparse networks, forwarding chains are likely broken due to limited availability of one-hop routes.

The index structure used in our location lookup service is in fact virtual and is distributed across mobile hosts in a network. Each mobile host maintains a portion of this index structure. The whole index structure is integrated from the partial index structures maintained in individual hosts across a network. The construction of this decentralized index structure follows a *scalable distributed data structure (SDDS)* approach [10]. Duties of maintaining location information are distributed across hosts, and the situation that an individual host is much more heavily loaded than others is unlikely. Load balancing across hosts serves for making a network operated stably.

Evaluated by simulating mobile wireless networks, the performance of our location lookup service exhibits high success rates on location look-ups and low demands on network resources. The necessary condition for our location lookup service to work well is that mobile hosts are densely distributed in a network.

In the rest of this paper, we describe previous work relating to our work in Section 2. The description of the index structure is in Section 3. The method of location distribution and lookup is described in Section 4. The performance evaluation of this method is shown in Section 5. At last, we summarize the design of our location lookup service in Section 6.

2 Related Work

The table-driven and source-initiated on-demand ad hoc routing protocols in mobile wireless networks are surveyed in [17]. In ad hoc routing protocols making no use of location information, non-neglectable amount of channel bandwidth and node processing power are consumed in route discovery.

In order to reduce routing overhead in mobile wireless networks, Tsuchiya [19] first proposed the idea of establishing a landmark hierarchy. A landmark is typically a router which maintains routing information for other network devices in a scope. The landmark devices in different scopes interconnect themselves. Any packet sent between scopes goes through the corresponding pair of landmark devices. Thus, routing information is only maintained at and exchanged among the landmark devices. Gerla *et al.* [21, 4] proposed the LANMAR landmark routing protocol in wireless networks with group mobility. A group of clustered network devices moving together is treated as a scope, and one landmark device is elected in each scope. Only landmark devices maintain routing information. Location information is not explicitly used in the LANMAR protocol, but proximity among mobile hosts is made use of in forming scopes.

When location information is made use of, the overhead of ad hoc routing in mobile wireless networks could be greatly reduced. Ko *et al.* [8] proposed the Location-Aided Routing (LAR) protocol for ad hoc routing in mobile wireless networks. In LAR, no routing information needs to be explicitly maintained, instead, adjacency between mobile hosts is made use of. A packet is forwarded by a set of adjacent hosts, in sequence, to gradually reach its destination, and it is forwarded closer to its destination each time. Hong *et al.* [6] also showed that making use of geographic locations helps to reduce routing overhead in large-scale wireless networks.

Geographic forwarding of packets is generally used in location-based routing. Karp *et al.* [7] proposed the Greedy Perimeter Stateless Routing (GPSR) protocol in wireless networks. In this protocol, packets are forwarded to progressively move closer to their destinations. A next-hop node is picked up under the concern of greedily moving a packet geographically closer to its destination over a single-hop connectivity. Just because temporary connectivities among mobile hosts are made use of in forwarding packets, packets are not guaranteed to be forwarded to their final destinations. In order to make packets be likely forwarded to their final destinations, the Geocast routing protocol [11] as proposed to make a packet to be forwarded to multiple next-hop hosts for better chances of reaching its destination.

Geographic forwarding is also used in content delivery networks. Ratnasamy *et al.* [16] proposed the scalable Content-Addressable Network (CAN) in which content can be stored and retrieved by keys of content. A content space is an abstract multi-dimensional coordinate space, which is mapped to a set of hosts each of which manages one partition of the content space. In order to retrieve/update a content, a multi-dimensional coordinate is first derived from the key of this content; then using this coordinate, a retrieval/update request is geographically forwarded to the host holding this content.

Location lookup services are necessary to aid location-based routing in mobile wireless networks. Li *et al.* [9] designed the geographic forwarding based Grid Location Service (GLS). Each mobile host distributes replicas of its up-to-date location to a number of location servers across a geographic area, and a query about the location of a target host is served through geographically forwarding this query to a location server holding a replica of the location of the target host. A packet might take multiple geographic hops when it is forwarded from one location server to another one. The more times a packet is geographically forwarded, the more negative impact on data transmissions. When a channel is occupied in doing routing related transmissions, it is blocked from doing useful data transmissions [5].

Heavy overhead on forwarding packets has been a problem for geographic forwarding based protocols used in either mobile wireless networks or content delivery networks. Hence, matching the high-level forwarding topology to the topology of an underlying network has been taken into account of in recent studies. Tapestry [22] is a peer-to-peer overlay routing infrastructure for sending requests to servers nearby. A routing mesh is maintained in Tapestry for routing messages, which reflects the topology of an underlying network connectivities. Ratnasamy *et al.* [15] introduced the topology-aware CAN to take advantage of the topology of the underlying network to reduce overhead on forwarding packets. The application-level CAN topology is made to match the underlying network-level topology by clustering geographically proximate hosts together.

In our method, spatial adjacency between mobile hosts is made use of in forwarding packets. One hop in the forwarding chain of a service packet corresponds to a direct link between two adjacent hosts.

Decentralized index structure has the advantage of good scalability because no atomic operation is required across multiple entities, and it has been made use of in a number of application scenarios. Bozanis *et al.* [2] introduced the Logarithmic Dictionary Tree (LDT). Each entity in LDT only maintains its own view of a distributed environment. An operation is accomplished through a collaboration among a set of entities via forwarding it among them. Aberer [1] introduced a distributed and balanced binary search tree used for retrieving data objects. Each host only maintains a small portion of the whole set of data objects, and a retrieval about a data object is served through a collaboration among a small set of hosts with regard to the total number of data objects. Stoica *et al.* [18] proposed the *Chord* protocol for looking up content by their keys. The index structure used in *Chord* is organized into a balanced binary search tree. A content look-up, as well as an update on the index structure upon changes of content, only involves a small number of hosts. In GLS [9], a distributed and balanced search tree built on identifiers is used as the index structure. Using this index structure, a location update or query can be served through a cooperation among a small group of location servers.

In our method, a decentralized index structure is constructed into a distributed complete binary search tree. This index structure serves to define the cooperative relations among mobile hosts in serving location updates or queries, moreover, it also serves to limit the number of times a service packet is forwarded.

3 The Index Structure

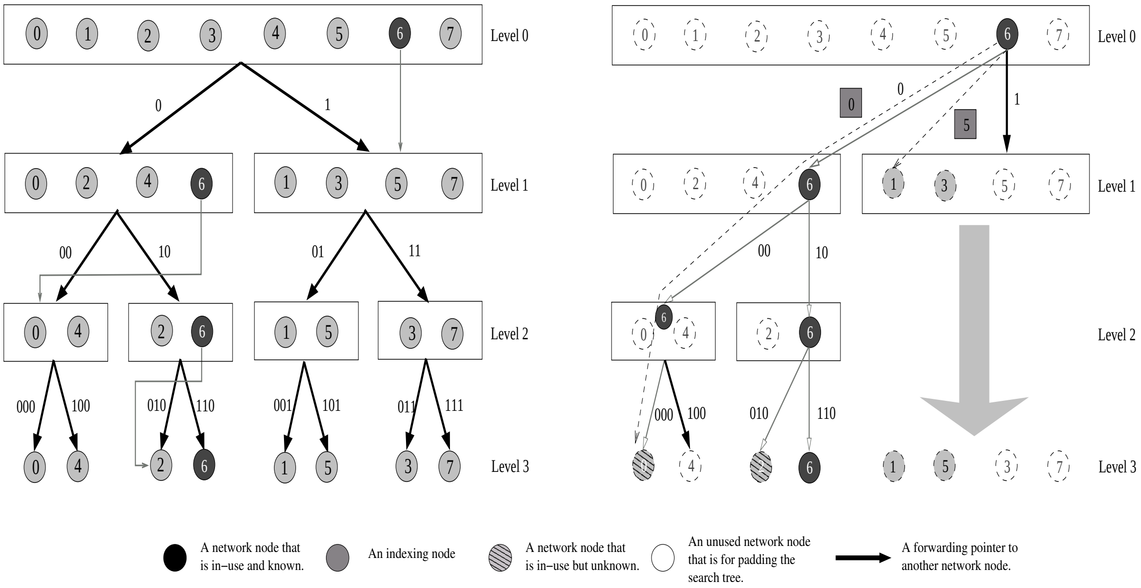
In our location lookup service, a complete binary search tree built on host identifiers is used as the index structure, which is constructed based on a relation of sharing common suffixes of host identifiers. In this search tree, leaf nodes represent host identifiers, and internal nodes represent groups of host identifiers sharing common suffixes of various lengths. Figure 1 shows a complete binary search tree built on identifiers of 8 mobile hosts, where each internal node is labeled with the common suffix of host identifiers included in it. In a network with K hosts, a complete binary search tree is of a height of $\lceil \log_2 K \rceil + 1$ (some leaf nodes may be empty if $\log_2 K$ is not an integer).

In the search tree, relations of sharing common suffixes of host identifiers can be formally expressed using modular arithmetics. When tree levels l are ascendant numbered from 0 at the root, a modular arithmetic by modulo 2^l could be defined in tree level l. Internal nodes in tree level l represent various equivalence classes formed in modular operations by modulo 2^l, and thus, an identifier is included in exactly one internal node in every tree level. Every internal node is forked into two child nodes which are equal partitions of their parent node.

Sharing various-length common suffixes characterizes extents of intimacy among friend hosts. The longer common suffix of identifiers is shared between a pair of hosts, the closer friends are them. When friendship relations are widely understood by every host in a network, they can be used in determining the cooperative relations among hosts in processing location queries or updates.

The search tree serves as a guideline for a host to pick up a next-hop host to forward a packet of location query or update to. The forwarding process corresponds to traversing the search tree along a path going from the root to a leaf node representing the target host of interest, *e.g.* the possible choices of a next-hop host which can be selected by host 6 is shown in Figure 1. Furthermore, the height of the search tree represents the maximum number of times a location query or update can be forwarded.

A host could hold location replicas for any of its friend hosts. Correspondingly, a host could distribute its location replicas to multiple friend hosts. The success rates of answering location queries are improved due to more chances of finding a location replica of a target host in a network, however, this benefit is achieved by paying a price of consuming more storage space and network bandwidth. For instance, in the search tree under the straight mode as illustrated in Figure 1-(b), host 6 could hold location replicas for any of its friend



(a) A Complete Search Tree (b) A Partial Search Tree at a Host

Figure 1: Serving location queries about host 6 using the search tree.

hosts, like holding location replicas for host 0 and 2.

It is only exemplary to construct the index structure into a complete binary search tree. Making the search tree balanced is the essential concern so that the height of the search tree grow logarithmically with the number of hosts in a network.

4 The Method of Location Distribution and Lookup

A location distribution and lookup method can be constructed by using a complete binary search tree built on host identifiers as the index structure. The following terms are used in the description of our method. A mobile host is called an *issuer* when it initiates a location query or update. A mobile host is called a *target* when its location is under query, or a packet of location update is being sent to it. A mobile host is called a *responder* when it could answer a location query about a target host.

4.1 Storage Space

Each mobile host maintains two data structures in its storage: a location table and a table of one-hop routes. A location table is used for holding location replicas for its friend hosts. A location is typically represented as a (x, y) pair in a geometric setting. A location can also be expressed in other formats depending on application scenarios. Every host also maintains a table of one-hop routes to reflect one-hop connectivities between itself and its current neighbors by keeping track of currently alive hosts in its neighborhood. No storage space is needed in each host to maintain the search tree.

4.2 Discovery of Single-hop Routes

Temporary single-hop connectivities between mobile hosts are made use of in forwarding location lookup related packets. Single-hop connectivities can be discovered through making each host periodically broadcast an alive message. Broadcasting alive messages does not consume much network bandwidth because the signal strength of a wireless transmission decays proportionally to the square of the distance from the transmitter [20]. Thus, an alive message can only reach those hosts which are currently in the neighborhood of the broadcasting host.

Upon receiving an alive message, a host treats the broadcasting host as its current neighbor and records a single-hop route between itself and the broadcasting host. A single-hop route expires after a certain time period. Similar to [9], the expiration interval is made twice the duration of an inter-broadcasting interval which relates to the moving speed of a host. When the moving speed of a host is above a threshold, it broadcasts an alive message whenever it has moved a certain distance; otherwise, it broadcasts alive messages at a pre-determined constant pace.

4.3 Updating on Locations

A mobile host distributes its up-to-date location to its friends for enhancing chances of answering queries about its location. A location update is different from an alive message in ways that: 1) an alive message is broadcasted, whereas, a location update is always unicasted to one host; 2) an alive message is never forwarded, whereas, a location update is very likely forwarded.

A mobile host periodically propagates its up-to-date location to its peer host, *i.e.* the two hosts share all bits of their identifiers except the highest bit. Every location update is issued only once by its issuer host, but it could be forwarded multiple times afterwards. Along a forwarding path, an intermediate host picks up a next-hop host from its current neighbors to forward a location update to, and an update on the location of a target host is forwarded to a closer friend of the target host each time. Upon receiving a location update, a host updates its location table with the location received, and the way the location table is updated depends on the mode of the search tree.

Under the straight mode, a mobile host can only distribute its location to its peer host. Thus, intermediate hosts along a path of forwarding a location update do not update their location tables. A host can not distribute its location anywhere if the forwarding chain is broken.

Under the enhanced mode, a mobile node could have its location distributed to any of its friend hosts. The intermediate hosts along a path of forwarding a location update can update their location tables with the location information received. Thus, even if a forwarding chain is broken, the issuer host can still distribute its location information to intermediate hosts up to the host where a forwarding chain is broken.

4.4 Looking Up Locations

Upon receiving a location query, a host first looks into its location table for an answer. On a hit, this mobile host responds the location of interest to the issuer host of this query; on a miss, it forwards the query to a next-hop host for further service. Similar to forwarding packets of location update, both friendship relations and underlying temporary connectives between hosts play roles in picking up a next-hop host. When a host can not answer a location query about a target host, it picks up a next-hop host in its current neighborhood so that the location query can be forwarded to an even closer friend of the target host. A location query will not be forwarded more times than the height of the search tree.

A location query fails when the forwarding chain is broken before it reaches a responder host. Upon a failure, a "*not found*" message is responded to the issuer host of a query by the host which last received this query.

4.5 Service Packets

Four types of packets are used in our method of location distribution and lookup. **Hello** packets are used for broadcasting aliveness of hosts. **Location Update** packets and **Location Query** packets are for updating and looking up locations of mobile hosts, respectively. Packets of **Response to A Location Query** are for carrying answers to location queries back to issuer hosts of location queries. The generic format of these four types of packets is shown in Table 1.

Packet Type
(ID, Location) pair of the issuer host
(ID, Location) pair of the target host
(ID, Location) pair of a next-hop host
Other location-related information

Table 1: The generic format of packets used in our method.

A host declares its aliveness through broadcasting an Hello packet. The host itself is specified as the issuer host and puts its own identifier and current location in an Hello packet. The other two (ID, location) pairs are left blank. A timeout value is specified in the other information field to invalidate the aliveness of a host when timer expires.

A host propagates its up-to-date location through sending a packet of location update to its peer host. The issuer host is the host initiating a location update, and the target host is the peer host of the issuer host. Only the identifier of the target host needs to be specified in a packet, and the location information of the target host is not needed. The (ID, location) pair of a next-hop host is overwritten each time by an intermediate host along a forwarding path. A timeout value is specified to invalidate the validity of the location being propagated when timer expires.

A host tries to discover the location of a target host by sending out a Location Query packet. The issuer host is the host initiating a location query, and the target host is the host whose location is under query. Only the identifier of the target host needs to be specified. A next-hop host is a host in the forwarding chain of moving a location query to a potential responder host. The (ID, location) pair of a next-hop host is also overwritten each time the packet is forwarded. In order to retrieve an up-to-date location of a target host, a last update time is specified in the packet to signify that a location replica of the target host is considered valid only when the replica was last updated no earlier than the time specified. A host responds to a location query through sending a packet of response to a location query. The issuer host and the target host of a packet of response to a location query is the responder host and the issuer host of a location query, respectively. When there is no next-hop host can be selected to further forward a location query to, a "not found" message is responded as an answer to the issuer host of this location query.

5 Performance Analysis

We evaluate the performance of our method of location distribution and lookup by simulating mobile wireless networks using the `ns-2` simulator [14]. Two metrics are used in the evaluation: success rates on answering location queries and demands on network resources. Demands on network resources are materialized into bandwidth consumption and the per-host storage occupation.

5.1 Simulation Scenarios

We simulate mobile wireless networks using the wireless and mobility extension [13] to the `ns-2` simulator. Each mobile node is equipped with an IEEE 802.11 wireless interface. Mobile hosts move within a $3000\mathrm{meter} \times 3000\mathrm{meter}$ square, and their movements follow a random waypoint model [3]. A mobile host moves along a straight line to a randomly chosen destination; upon arriving at the destination, it randomly chooses a new destination after pausing for 4 seconds. This process repeats till a simulation finishes. In all mobility scenarios, the maximum movement speed is 30 meters/second, and average movement speeds range between 5.4 and 6.1 meters/second. Each mobile host performs constant bit-rate (`cbr`) data transmissions. All simulations last 300 seconds.

5.2 Performance of Our Method of Location Distribution and Lookup

We demonstrate the performance of our method using scaling behaviors of evaluation metrics, and we also present the advantage of our method by comparing the performance of our method to the one of GLS [9] under the same set of simulation scenarios. When the search tree runs in the enhanced mode, the performance of our method is shown in Figure 2. The performance of GLS is shown in Figure 3.

5.2.1 Success Rates on Answering Location Queries

A location query is responded with either the up-to-date location of the host under query, or a "not found" message upon a failure. A success rate on answering location queries is the ratio of the number of successful location retrievals to the total number of location queries issued. Figure 2-(a) shows the scaling behavior of success rates. Success rates are high when there are more than 100 hosts in a network. This fact signifies that forwarding chains are unlikely broken when each mobile host is surrounded by more and more neighbors in a network, as shown in Figure 2-(b). Compared to scenarios where hosts are sparsely distributed in a network, *e.g.* 16 hosts, forwarding chains are often broken.

With regard to success rates on location discovery in GLS, shown in Figure 3-(a), our method has higher success rates. This fact states that hosts driven under our method cooperate more efficiently.

5.2.2 Demands on Network Resources

The average per-host bandwidth consumed in our method is shown in Figure 2-(c) and 2-(d). The per-host bandwidth consumption goes up with higher densities of mobile hosts in a network. On one hand, the higher density of hosts, the higher bandwidth consumption on broadcasting aliveness messages. On the other hand, when one-hop connectivities between mobile hosts become more available as densities of hosts go up (see Figure 2-(b)), forwarding chains are unlikely broken, in turn, more bandwidth is consumed in location distribution and lookup. In the meantime, the ratios of the amount of bandwidth consumed in our method to the one consumed in data transmissions stay roughly at the same level, independent to densities of mobile hosts in a network. That is, only a small and constant portion of the bandwidth is consumed under our method. With regard to bandwidth consumptions in GLS, shown in Figure 3-(c), our method consumes less amount of bandwidth. Both GLS and our method consume a small and constant portion of the amount of bandwidth available in a network.

The number of times a service packet is forwarded highly contributes to bandwidth consumptions in a location service. As shown in Figure 2-(e), service packets of various types in our method are forwarded only once, on average. Whereas, service packets in GLS are geographically forwarded more times, shown in Figure 3-(e), for a reason that one forwarding between a pair of location servers takes multiple geographic hops [9].

The per-host storage occupation for holding location replicas is evaluated by the ratio of the number of location replicas held in each host to the total number of mobile hosts. The maximum and average per-host storage occupation in our method are shown in Figure 2-(f). In most scenarios, each mobile host only needs to hold location replicas for less than 10% of other hosts. In some scenarios, per-host demands on storage space are higher, but they are no more than 25%. With regard to the per-host storage occupation in GLS as shown in Figure 3-(f), both our method and GLS do not occupy much storage space in each host for holding location replicas.

6 Conclusions

We described a method of location distribution and lookup, which aims to restrict the overall overhead in location discovery needed by location-aided routing. The design goal of this method is to have high success rates on answering location queries while keeping demands on network resources low. In order to achieve this goal, a complete binary search tree built on host identifiers is used as a decentralized index structure which guides each location query or update be served via a cooperation among a small number of mobile hosts.

Meanwhile, transient one-hop routes between mobile hosts are made use of in selecting cooperative hosts.

In order to achieve high success rates on answering location queries, it is important for each host to distribute replicas of its location to other hosts. In order to restrict demands on network resources, location updates are propagated through a sequence of single-hop route only to a small set of other hosts.

By simulating mobile wireless networks, the performance of this method exhibits high success rates on answering location queries and low consumption on bandwidth and per-host storage. Compared to the performance of GLS, our method exhibits higher success rates and lower demands on network resources. A prominent feature of our method is that decisions on forwarding service packets are jointly determined by the structure of the embedded search tree and the temporary connectivities in the network.

References

[1] Karl Aberer. P-Grid: A self-organizing access structure for P2P information systems. *Lecture Notes in Computer Science*, 2172:179 – 192, 2001.

[2] Panayiotis Bozanis and Yannis Manolopoulos. Ldt: a logarithmic distributed search tree. In *Proceedings of the 4th Workshop on Distributed Data and Structures (WDAS'02)*, Paris, France, 2002.

[3] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of ACM/IEEE Conference on Mobile Computing and Networking (MobiCom)*, pages 85–97, October 1998.

[4] Mario Gerla, Xiaoyan Hong, and Guangyu Pei. Landmark routing for large ad hoc wireless networks. In *Proceedings of IEEE GLOBECOM 2000*, November 2000.

[5] Piyush Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transations on Information Theory*, 46(2):388 – 404, March 2000.

[6] Xiaoyan Hong, Kaixin Xu, and Mario Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network Magazine*, pages 11–21, July–August 2002.

[7] Brad Karp and H. T. Kung. Greedy perimeter stateless routing for wireless networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 243–254, Boston, MA, August 2000.

[8] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (lar) in mobile ad-hoc networks. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM'98)*, pages 66 – 75, Dallas, Texas, 1998.

[9] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the*

6th ACM International Conference on Mobile Computing and Networking (MobiCom '00), pages 120 – 130, Boston, Massachusetts, August 2000.

[10] Witold Litwin, Marie-Anne Neimat, and Donovan A. Schneider. Lh: Linear hashing for distributed files. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 327 – 336. ACM Press, 1993.

[11] Christian Maihfer. A survey of geocast routing protocols. *IEEE Communications Surveys & Tutorials*, April 2004.

[12] Dragos Niculescu and Badri Nath. Ad hoc positioning system (APS) using AoA. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, CA., 2003.

[13] The Rice Monarch Project. The Wireless and Mobility Extensions to ns-2. Available from http://www.monarch.cs.cmu.edu/. Computer Science Department, Rice University.

[14] The VINT Project. The network simulator - ns-2. Available from http://www.isi.edu/nsnam/ns/. UC Berkeley, LBL, USC/ISI, and Xerox PARC.

[15] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *Proceedings of IEEE INFOCOM'02*, 6 2002.

[16] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content addressable network. In *Proceedings of ACM SIGCOMM*, 2001.

[17] E. Royer and C. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. In *IEEE Personal Communications*, April 1999.

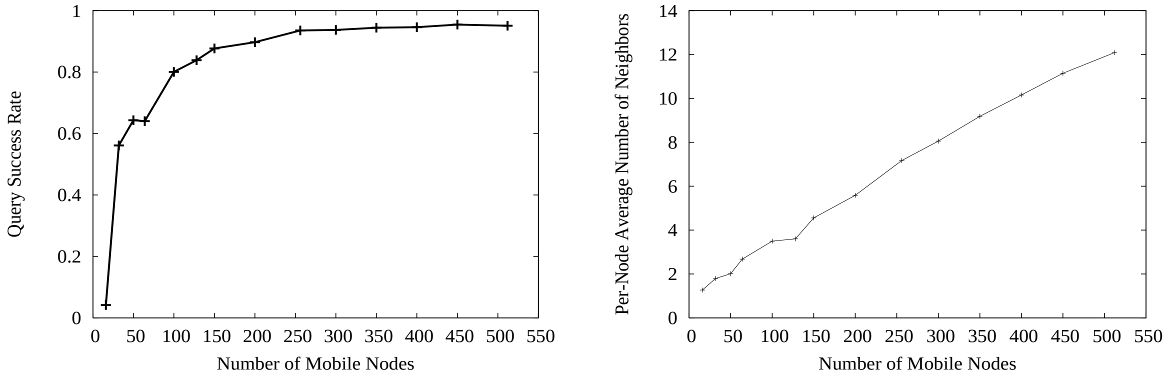
[18] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149 – 160, 2001.

[19] P. F. Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. In *Symposium proceedings on Communications architectures and protocols*, pages 35–42. ACM Press, 1988.

[20] H. S. Wang and P. C. Chang. On verifying the first-order markovian assumption for a rayleigh fading channel model. In *IEEE Transaction on Vehicular Technology*, volume 45, pages 353–357, May 1996.

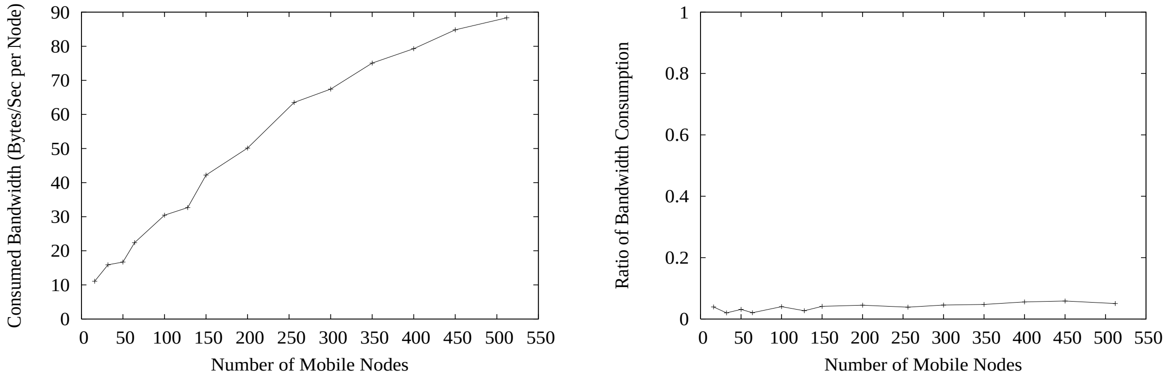
[21] Kaixin Xu, Xiaoyan Hong, and Mario Gerla. Landmark routing in ad hoc networks with mobile backbones. *J. Parallel Distrib. Comput.*, 63(2):110–122, 2003.

[22] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1), January 2004.



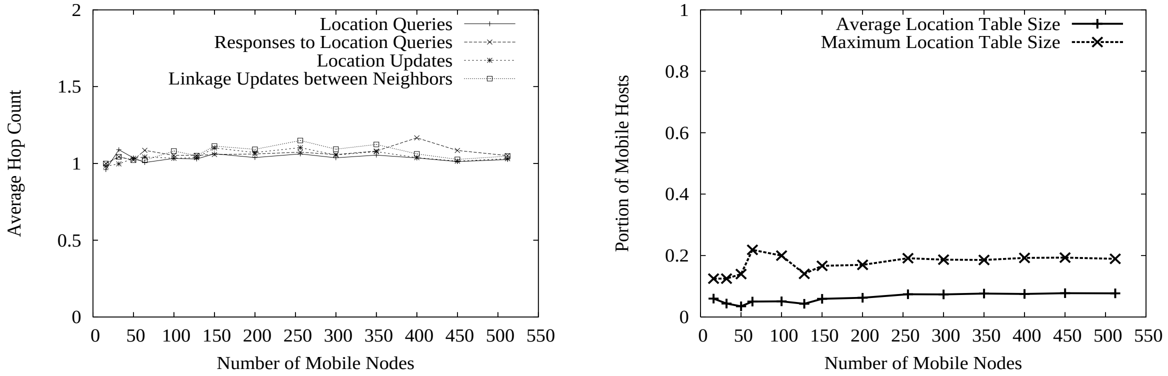
(a) Success Rates on Answering Location Queries

(b) Average Number of Neighbors



(c) Bandwidth Consumption

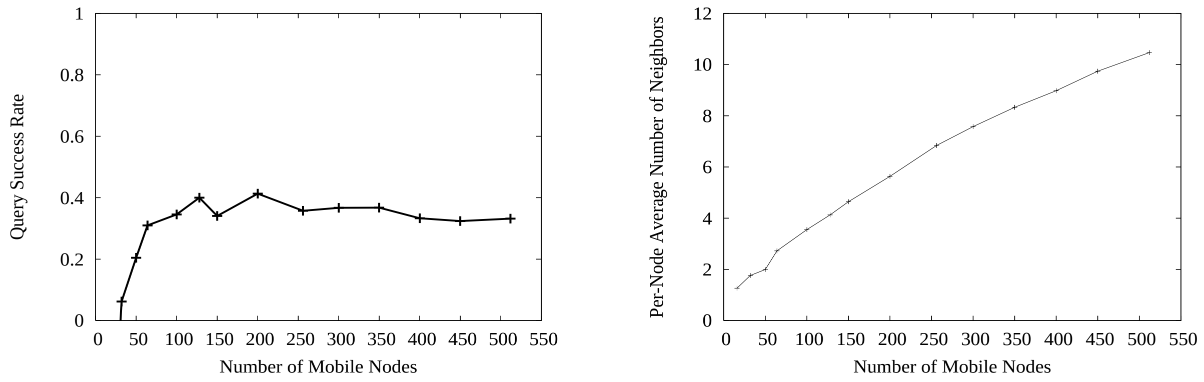
(d) Ratio of Bandwidth Consumption



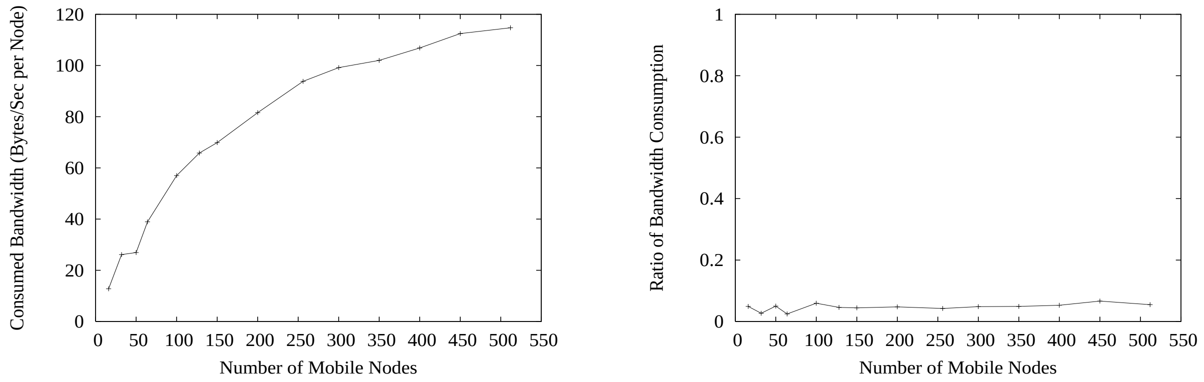
(e) Average Length of Forwarding Paths

(f) Storage Occupation

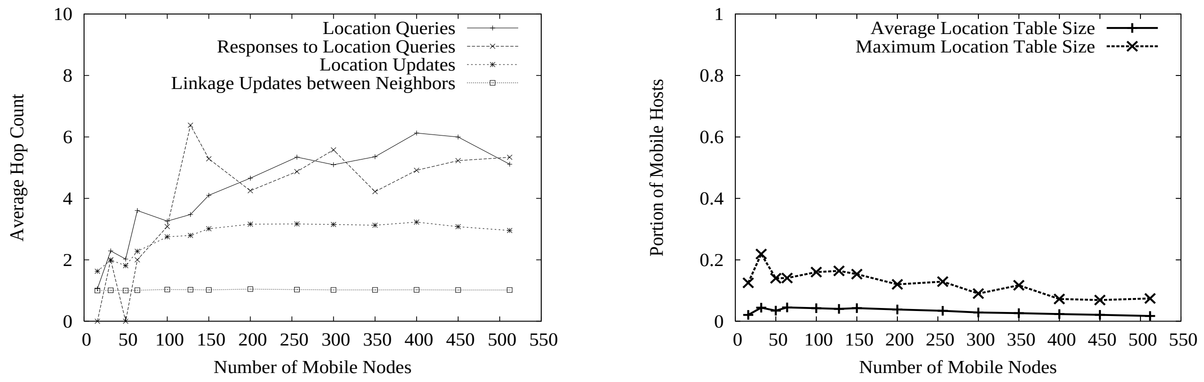
Figure 2: The performance of our method of location distribution and lookup. The search tree is under the *enhanced* mode. Simulations are run in a $3000\mathrm{m}eter \times 3000\mathrm{m}eter$ square



(a) Success Rates on Answering Location Queries (b) Average Number of Neighbors



(c) Bandwidth Consumption (d) Ratio of Bandwidth Consumption



(e) Average Length of Forwarding Paths (f) Storage Occupation

Figure 3: The performance of GLS. Simulations are run in a $3000meter \times 3000meter$ square.