4-26-2010

A Peer-Tree Based Location Lookup Service in Mobile Wireless Networks

Jun Liu
University of North Dakota, jun.liu@UND.edu

Recommended Citation
Jun Liu. "A Peer-Tree Based Location Lookup Service in Mobile Wireless Networks" (2010). *Computer Science Faculty Publications*. 25.
https://commons.und.edu/cs-fac/25

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 593107, 10 pages
doi:10.1155/2010/593107

Research Article

A Peer-Tree Based Location Lookup Service in Mobile Wireless Networks

Jun Liu

Computer Science Department, University of North Dakota, Grand Forks, ND 582029015, USA

Correspondence should be addressed to Jun Liu, junliu.mail@gmail.com

Received 9 September 2009; Revised 16 February 2010; Accepted 13 March 2010

Academic Editor: Weihua Zhuang

We present a location look-up service for aiding location-based routing in wireless networks with mobile hosts. This location look up service is constructed based on a peer-tree indexing structure which represents the friendship relations among mobile hosts. The friendship is derived from the relations of sharing various-length common suffixes of host identifiers. A friendship relation determines the cooperation relation among mobile hosts. The Peer-Tree indexing structure provides a guideline for propagating location updates and queries in an organized manner. Each host periodically propagates its up-to-date location to a small number of its friends. Location queries are forwarded to and serviced at one of the hosts which hold a copy of the location under query. Paths for forwarding location updates or queries are jointly determined by the Peer-Tree indexing structure and the temporary connectivity in a network. The goals of this location look-up service include achieving a high success rate on location discovery, having low demands on network resources, and being robust to frequent changes of network topology.

1. Introduction

The location-based routing protocols determine the packet forwarding paths in wireless networks with mobile hosts. These protocols, for example, the Greedy Perimeter Stateless Routing (GPSR) protocol [1], do not make an explicit use of the routing tables because of the high overhead associated with maintaining the routing tables. With mobility, routing tables have to be frequently updated to reflect the frequent changes of network topology. The location-based routing protocols make a joint use of host identifiers and locations for determining the forwarding paths. The location-based forwarding makes each packet to stay geographically closer to the destination of the packet after each forwarding.

A location lookup service is expected to answer location queries with high success rates and to constrain the overhead on maintaining the location information. However, the two expectations often conflict to each other. On one hand, achieving high rates of successfully answering the location queries requires each host to propagate its current location to as many hosts as possible. On the other hand, frequent propagations of locations consume considerable amount of network bandwidth. A good balance needs to be maintained between propagating location updates and the consumption of network bandwidth.

In this paper, we describe a peer-tree based Location lookup Service (PTLS) in wireless networks consisting of mobile hosts. This service follows a decentralized approach by distributing the duty of maintaining the up-to-date locations to all participating hosts. This service aims to achieve high rates of successfully answering location queries, to constrain the consumption on network bandwidth, and to be highly available under frequent changes of network topology. This location lookup service is constructed based on a peer-tree indexing structure [2]. The peer-tree indexing structure is a complete binary tree which represents the friendship among hosts. Two hosts are friends if a common suffix is shared between their identifiers. The length of a common suffix of the identifiers shared between two wireless hosts denotes the degree of their friendship. The indexing structure used in the PTLS service is in fact virtual and is distributed across all participating mobile hosts in a network. Each mobile host maintains a part of this indexing structure, and the entire indexing structure is integrated from the partial indexing structures. The construction of this decentralized index structure follows a *scalable distributed* *data structure (SDDS)* approach [3]. Duties of maintaining location information are distributed across all participating mobile hosts, and the situation that an individual host is significantly more loaded than others is unlikely. Balancing the load across hosts facilitates the stable operation of the location lookup service.

Guided by the peer-tree indexing structure, the location updates and location queries can be propagated in an organized manner. Namely, location updates are propagated to a small number of other hosts, while location queries are forwarded to the hosts where the queries are likely answered. A location update and a location query are always forwarded to the closer friends of the host whose location is being updated and under query, respectively. The process of forwarding a location update or a location query can be imagined as traversing a path in the complete binary tree beginning at the root and moving toward the leaf level. Forwarding a location update or a location query is gradually forwarded to its closer friends, and the number of times a location update or a location query can be forwarded is restricted by the height of the peer-tree. An actual forwarding path is jointly determined by the peer-tree and the temporary connectivity in a network.

High rates on successfully answering location queries can be attained by using the peer-tree as a guideline for forwarding location updates and queries in an organized manner. When a host updates its location, it propagates its up-to-date location to a small group of its friend hosts. The group of friends which are chosen to receive the location update is jointly determined by both the indexing structure and the current network topology. The way that the location updates are propagated determines that the location queries should be forwarded to the hosts which likely hold answers. When querying about the location of a host, the location query is forwarded to one of the friends of the host whose location is under query. The query about the location of a target host fails to be answered when it can no longer be forwarded to the closer friends of the target host.

The bandwidth consumption can be constrained when location updates and queries are forwarded under the guidance of the peer-tree indexing structure. When a wireless host updates its location, its location update is only sent to a small number of its friends. Likewise, a location query is always forwarded to a small group of friends of the host under query because an answer is likely to be found at these friends.

Robustness to changes of network topology can also be achieved by distributing location information about a target host onto multiple wireless hosts. Thus, a location query can be answered by any of the hosts at where an answer can be found. Since each wireless host periodically propagates its current location, its up-to-date location information can be propagated to many of its friends over time if it does not move too fast. Hence, the success rate of answering location queries is not affected by the frequent changes of network topology to a great extent.

Evaluated by simulating wireless networks consisting of mobile hosts, the performance of the peer-tree based location lookup service exhibits high rates of successfully answering location queries, a low consumption on network bandwidth, and robustness to changes of network topology. The necessary condition for this location lookup service to work well is that the density of hosts in a network is reasonably high, such that it is unlikely to have a cluster of hosts to be isolated from the rest for an extended time duration.

The rest of the paper is organized as follows. The previous work relating to our work is described in Section 2. The description of the peer-tree indexing structure is described in Section 3. The PTLS service is described in Section 4. The performance evaluation of the PTLS service is demonstrated in Section 5. The PTLS service is summarized in Section 6.

2. Related Work

2.1. Locationing Techniques for Mobile Stations in Wireless Networks. Most of the localization technologies in the wireless networks are landmark based. In landmark-based localizations, a number of fixed access points (APs) periodically broadcast their accurate positions. A mobile host could determine its current location through geometric computations using the estimated distances between a mobile station (MS) and the fixed APs. The distances are estimated through various measurements obtained from radio frequency signals that are transmitted between the MS and the fixed APs. The measurements can be categorized into the received signal strength-based metrics and the time delay-based metrics.

The received signal strength (RSS) [4] exploits the relation between power loss of the signals and the distance [5]. RSS is defined as the voltage measured by a receiver's received signal strength indicator (RSSI) circuit [6]. RSS measurements are relatively inexpensive and simple to implement in hardware, though they are easily affected by two sources of error: multipath signals and shadowing [6].

Time delay-based metrics form the finer resolution distance measurements by estimating the propagation time of the line-of-sight (LOS) signals. The time delay-based metrics include time of arrival (TOA), time difference of arrival (TDOA), and round-trip time of arrival (RTOA). An TOA measurement is defined as the time of transmission plus a propagation-induced time delay [7]. TOA measurements require a common time reference between wireless stations, whereas TDOA and RTOA measurements can be obtained in an asynchronous environment. Time delay-based signal metrics are susceptible to two sources of error: additive noise and multipath signals.

The angle of arrival (AOA) measurements [8] are defined as the direction to neighboring sensors (rather than the distance to neighboring sensors). The AOA measurements require each wireless station to be equipped with two (or more) directional antennas that point in different directions. The AOA measurements can be incurred in two approaches. The most common method is to use an array of wireless stations which adopt the array signal processing techniques. AOA is estimated from the differences in arrival times for a transmitted signal at each wireless station in the array. The second approach is to estimate the RSS ratio between two (or more) directional antennas that are located on a wireless station. An AOA measurement is estimated from the ratio of the individual RSS values at two directional antennas.

2.2. Traditional Location-Ignorant Routing Protocols. Traditional ad hoc routing protocols maintain routing tables in order to determine the paths for forwarding packets [9]. Each host proactively maintains its routing table to reflect its current view of the network topology. Each wireless host propagates the content of its routing table to others either periodically or upon content changes. A wireless host updates its own routing table based on the content of a routing table that is propagated by another host, such as the Destination-Sequenced Distance-Vector (DSDV) routing protocol [10]. Route maintenance is performed by each host even when there is no pending data transmissions.

2.3. Location-Aided Routing Protocols. In order to restrict the overhead in maintaining routing tables, some ad hoc routing protocols have been proposed to eliminate or restrict the use of routing tables. Landmark-based routing protocols [11] only make limited use of routing tables. A landmark hierarchy is established in a landmark-based routing protocol. A network is divided into scopes. Each scope maintains one landmark, typically a router, which maintains routing information within the scope. The landmarks in different scopes interconnect themselves. Any packet sent between different scopes goes through the corresponding pair of landmark hosts. Landmarks exchange routing information among themselves. Other landmark-based routing protocols include the LANMAR routing protocol [12] and the Location-Aided Routing (LAR) protocol [13].

Geographically forwarding packets is generally used in location-based routing protocols, such as the GPSR protocol [1]. Geographic forwarding is also used in content delivery networks. The scalable Content-Addressable Network (CAN) [14] also makes use of the geographic forwarding. The key of a content is used as the storage location of the content. Queries to a content are geographically forwarded to the corresponding storage location whose address is the key of the content.

2.4. Indexing Structures. Decentralized index structures have been used in many application scenarios. In the Logarithmic Dictionary Tree (LDT) [15], each entity maintains its own view of an entire distributed tree. An operation is accomplished through the collaboration of a set of entities by forwarding it among these entities. The *Chord* protocol [16] is a content retrieval service by the content keys. An indexing structure is used in *Chord* and is organized into a balanced binary search tree. Both content lookups and updates to content only involves a small number of hosts. Tapestry [17] is a peer-to-peer overlay routing infrastructure for sending requests to nearby servers. A decentralized indexing structure provides good scalability because no atomic operation is required across multiple entities.

Other decentralized indexing structures include PGrid [18], Chord [16], and peer-tree [2]. PGrid is a distributed and balanced binary search tree that is 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. Chord is a balanced binary search tree. A content lookup or a content update only involve a small number of hosts. peer-tree is constructed as a distributed complete binary search tree. The peer-tree structure aims to limit the number of times that a service packet is forwarded by defining the cooperative relations among entities.

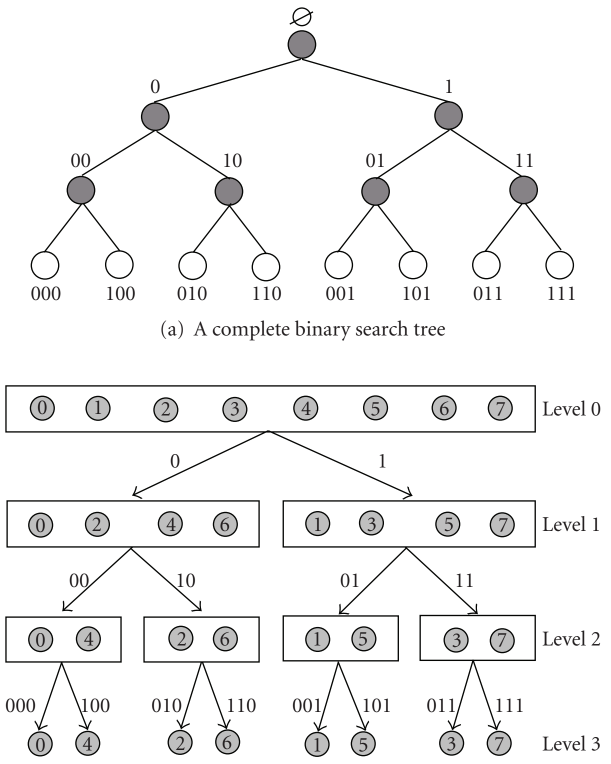
2.5. Comparison of the PTLS Location Lookup Service with Other Location Lookup Services. The PTLS service has two prominent properties. First, PTLS restrains the bandwidth consumption. This is achieved by making one forwarding hop at the protocol level to correspond to one hop of forwarding in the underlying physical network. Second, PTLS forwards location updates and queries along multiple paths. Multi-path forwarding serves to achieve both the high rates of successfully answering location queries and the robustness to changes of network topology.

The Grid Location Service (GLS) [19] is a location lookup service based on geographically forwarding location updates and queries. Each mobile host distributes its up-to-date location information to a number of location servers across a geographic area. A location query about a target host is served through being geographically forwarded to a location server which holds an answer. Heavy overhead on forwarding packets has been a problem in GLS. One hop of packet forwarding at the protocol level may correspond to multiple hops in the underlying physical network. Without taking into account the connectivity in an underlying network, a large amount of network bandwidth can be consumed in forwarding packets. In the topology-aware Content-Addressable Networks [20], a routing mesh is made to match the underlying network-level connectivity by clustering geographically proximate hosts.

Multi-path forwarding has also been adopted in the wireless ad hoc routing protocols. In the DSR protocol [21], a source host can maintain multiple routes to a destination under highly dynamic network connectivity. The Geocast routing protocol [22] forwards a packet through multiple routes in order to enhance the chances of reaching a destination under dynamic network connectivity. The service-restorable bandwidth-guaranteed routing [23] maintains two routing paths. One is the main path, and the other one is the backup path. Efficient path selection algorithms are desired constrain the computational overhead in arranging the multipath routes. The peer-tree structure adopted in the PTLS service facilitates the computationally efficient selection of paths.

3. The Peer-Tree Indexing Structure

The peer-tree indexing structure is constructed based on the relations of sharing common suffixes of host identifiers. These relations can be expressed into a complete binary search tree as shown in Figure 1(a). In the complete binary tree, each leaf node represents a unique host identifier, and each internal tree node represents a set of host identifiers which share a common suffix. An example peer-tree built on 8 host identifiers is shown in Figure 1(b).



(a) A complete binary search tree

(b) A peer-tree

Figure 1: Illustration of a peer-tree built on identifiers of 8 mobile wireless hosts. Each leaf node represents the identifier of a host, and each internal node represents a set of identifiers sharing a common suffix. Each internal node is labeled by the common suffix of identifiers included in it.

3.1. The Friendship Relations. Sharing common suffixes of various lengths reveals the degrees of friendship among hosts. Two hosts are closer friends if they share a longer common suffix between their identifiers. The friendship determines a cooperative relation among hosts. The friendship of different degrees can be expressed as the equivalence classes using the modulo operations. For a common suffix s with its length of $|s|$, an equivalent class of hosts is defined as $\{a : s \equiv a \mod 2^{|s|}\}$, where a denotes the host identifiers. The equivalent classes can be represented as the internal tree node in a complete binary search tree. The tree levels of such a search tree are incremented as it moves toward the leaf level with the root level being denoted as level 0. The leaf level is denoted as level L which corresponds to the height of the tree. The value of L is equivalent to the total number of bits of the identifiers. The depth of an intermediate tree level is denoted by l $(0 < l < L)$. When each internal tree node at tree level l is labeled by s $(0 \leq s \leq 2^l - 1)$, a node s represents an equivalence class: $E_l(s) = \{a : s \equiv a \mod 2^l\}$. A pair of hosts are the closest friends (a.k.a. *peers*) to each other if their identifiers share the longest suffix, that is, their identifiers only differs in the most significant bit, for example, two hosts with their identifiers a and b are peers if $a \equiv b \mod 2^{(L-1)}$.

The peer-tree indexing structure can serve as a guideline for forwarding location updates and queries. When a *target host* represents a host whose location is under query or is being updated, a location query or update is always forwarded to the closer friends of the target host. A closer friend of the target host should qualify two conditions. First, the closer friend is one of the neighbors surrounding the host which currently holds a location update or query. Second, the identifier of the closer friend must share the longest suffix with the identifier of the target host. The selection of a closer friend can be imagined as searching in the peer-tree. The search starts at the root and moves toward the leaf node representing the identifier of the target host. The height of a peer-tree represents the maximum number of times that a location update or query can be forwarded.

Although the peer-tree structure seemingly shares similarity with PGrid [18], they have significant differences. First, each entity in PGrid is only responsible for a particular set of data. In peer-tree, a host can hold location information for any of its friends on a dynamic basis. Second, the forwarding chain in PGrid is fixed. In peer-tree, the forwarding chain is jointly determined by the peer-tree structure and the dynamic network connectivity. Third, the main concern of PGrid is load balancing. Thus, a query may have to be forwarded to a particular host for a service. Moreover, the forwarding path cannot be flexibly determined under the dynamic network connectivity. In peer-tree, the main concern is to enhance the success rates of answering location queries while restricting the bandwidth consumption. Due to the opportunistic communications among hosts, it is ideal that a location update or query can be flexibly serviced by multiple hosts. Since PGrid does not provide such flexibility, it is not suitable for being used to form an indexing structure for aiding the forwarding of queries or updates in wireless networks with mobile hosts. The peer-tree indexing structure allows a flexibility based on a well-defined cooperative relation among hosts. The actual forwarding path is jointly determined by the set of eligible next-hop hosts and the dynamic network connectivity.

3.2. Determination of Paths for Forwarding Updates and Queries. The host that currently holds a location update or a location query is called the *current host*. A current host first locates the smallest equivalent class that includes its own identifier and the identifier of the target host. Then, a next-hop host is selected within this equivalent class based on two criteria:

Criterion 1. The identifier of the next-hop host should share the longest suffix with the identifier of the target host among the candidates included in the same equivalent class.

Criterion 2. The next-hop host has to be among the neighborhood of the current host.

The procedure of selecting a next-hop host is shown in Algorithm 1. The selection of a next-hop host corresponds to moving downward in the peer-tree. The traversal always starts at the root of the tree and ends at the smallest equivalent class which includes both the next-hop host and the target host. For example, the paths of forwarding location

```
[H] Input: Identidier $i$ of the current host, Identifier $j$ of
  a target host.
  Output: Identifier $k$ of a next-hop host.
CurrLongestSuffixLength ← SuffixLength($i$, $j$);
NextHopNode ← NULL;
foreach Host Identifier k in Neighbors[i] do
    if (CurrLongestSuffixLength < SuffixLength ($j$, $k$)) then
        CurrLongestSuffixLength ← SuffixLength($j$, $k$);
        NextHopNode ← $k$;
    end
end
SuffixLength($i$, $j$)
begin
    M ← 1;
    Length ← −1;
    $m_i$ ← 1;
    $m_j$ ← 1;
    while $m_i = m_j$ do
        Length ← Length + 1;
        M ← M · 2;
        $m_i$ ← i mod M;
        $m_j$ ← $j$ mod M;
    end
    return Length;
end
```

Algorithm 1: FindNextHopHost.

```
[H] Input: A location update $U$, Indentidier $i$ of the current
  host, Identifier $j$ of the host which initiated the update $U$.
NextHopHost ← FindNextHopHost($i$, $j$);
if NextHopHost ≠ NULL then
    Memorize the Location of Host $j$ that is included in $U$;
    Send Location Update $U$ to NextHopHost;
else
    Delete Location Update $U$;
end
```

Algorithm 2: ForwardLocationUpdate.

updates and queries made by host 6 are shown in Figure 2. Based on Algorithm 1, the operations of propagating location update and forwarding location query are defined as follows.

(1) *Propagating Location Updates*. When a host updates its location, it first sends the update to one of its friends that are in its current neighborhood. Then, this update can be further forwarded to the closer friends of the host which has initiated this update. At each forwarding, a location update is only unicasted to the next-hop host. The location information of the host whose location is being updated is memorized by each host along the forwarding path of a location update. Therefore, a location query about a target host could be answered by any of the friend hosts of this target host. Moreover, a host can hold location replicas for any of its friend hosts. The procedure of forwarding a location update is illustrated in Algorithm 2. For example, the updates of the location about host 6 can be replicated at hosts 0, 2, and 4 (ref. Figure 2(b)).

```
[H] Input: A location query $Q$, Indentidier $i$ of the current
  host, Identifier $j$ of a target host.
NextHopHost ← FindNextHopHost($i$, $j$);
if NextHopHost ≠ NULL then
    Send Location Query $Q$ to NextHopHost;
else
    Send the Response “Failed in Servicing Location Query”
      to Host $j$;
    Delete Location Query $Q$;
end
```

Algorithm 3: ForwardLocationQuery.

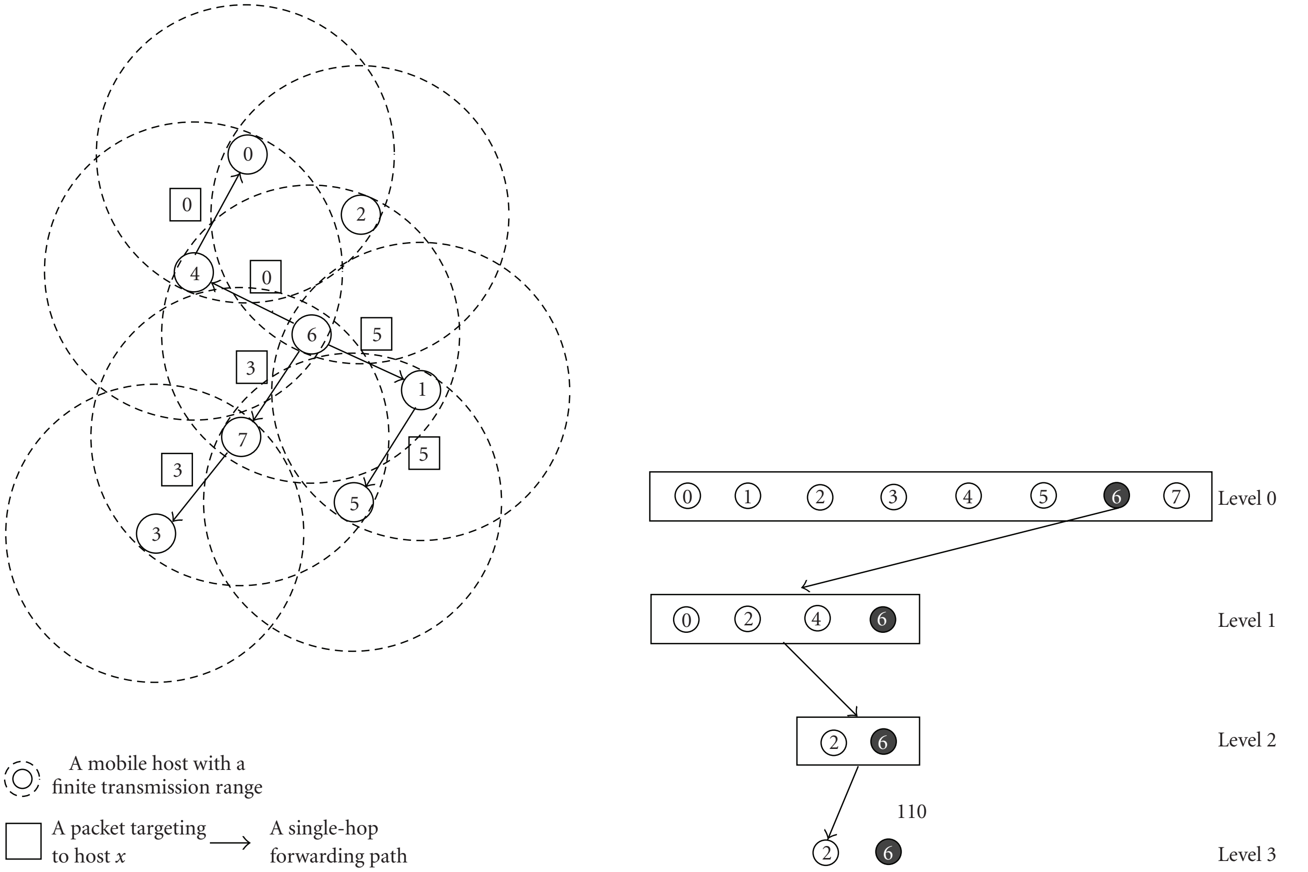
(2) *Forwarding Location Queries*. When the location of a target host with its identifier j is being queried, the location query is gradually forwarded to the closer friends of the target host. Assuming that a location query is currently in the hand of a host with its identifier i, but host i cannot answer this query. Host i has to forward this query to a next-hop host. Again, host i searches for a closest friend of the target host among its current neighbors. The procedure of searching a next-hop host for forwarding a location query can be illustrated in Algorithm 3. For example, the paths of forwarding location updates and queries made by host 6 are shown in Figure 2(a).

4. The Location Lookup Service

The main functionality of the peer-tree based location lookup service includes detecting direct neighbors, updating location information, and answering location queries. Four types of packets are used in our location lookup service. *Hello* packets are used for broadcasting the aliveness of hosts. *Location Update* packets are used for updating the locations of hosts. *Location Query* packets are used for looking up locations of hosts. *Response to A Location Query* packets are used for sending answers to location queries back to the issuers of location queries.

4.1. Relationship to the Location-Aided Routing Protocols and the Localization Subsystem. It is noteworthy that the PTLS service only facilitates the location-aided routing protocols, rather than acting as a routing protocol. The functions of the PTLS service is not dependent on an underlying routing protocol. The forwarding of PTLS-specific packets is jointly determined by the peer-tree indexing structure and the direct connectivity in the neighborhood.

The PTLS service runs on top of an underlying localization subsystem and disseminates the estimated locations of mobile hosts. A mobile host is assumed to possess the ability of estimating its current location. The accuracy of the localization is orthogonal to the functions of the PTLS service, though inaccurate location estimations are disseminated by


0
2
4
6
1
7
5
3
0
0
5
3
5
3
A mobile host with a finite transmission range
A packet targeting to host x
A single-hop forwarding path
0 1 2 3 4 5 6 7
Level 0
0 2 4 6
Level 1
2 6
Level 2
110
2 6
Level 3

(a) The network connectivity

(b) Forwarding paths

Figure 2: The paths for forwarding location queries made by host 6.

the PTLS service under inaccurate locationing. The functions of the PTLS service make no specific requirement on the format of locations. For simplicity, the format of locations used in this paper is assumed to be either a (x, y) pair in a 2-D geometric setting, or a (x, y, z) tuple in a 3D geometric setting with z representing the altitude.

4.2. Detecting the Temporary Single-Hop Connectivity. Each wireless host detects the temporary single-hop connectivity between itself and its neighbors by listening to the packets received from the wireless channel. Each host declares its existence by periodically broadcasting *Hello* packets. Due to the fast decay of the signal strength, a *Hello* packet can only be received by the hosts which are within the coverage of the sender. The following information is specified in a *Hello* packet: the broadcaster's identifier and current location, as well as a timeout value. A timeout value signifies the duration during which a single-hop connectivity is valid between the broadcaster and the receiver of a *Hello* packet. A timeout value is set as twice the interval between two consecutive broadcasting of *Hello* packets. A *Hello* packet is broadcasted once every 10 seconds of silence.

4.3. Updating Locations. Each mobile host propagates its up-to-date location to its friends through sending *Location Update* packet. A *Location Update* packet is different from a *Hello* packet due to the following two reasons. First, a *Hello* packet is broadcasted, while a *Location Update* is unicasted to a next-hop host. Second, a *Hello* packet is never forwarded, while a *Location Update* is forwarded if a next-hop host can be found. A *Location Update* packet includes the following information:

(i) the identifier and the location of the target host whose location is being updated,

(ii) the time when the location update is initiated by the target host,

(iii) a timeout value which specifies the valid duration of the location.

The frequency of updating the location is dependent on the speed of the movement of a mobile host. A mobile host updates its location once a certain distance has been traversed. The distance is related to the coverage of the signal strength of a mobile host. For example, a distance of 100 meters is appropriate for a coverage with a 250-meter radius.

The location information is gradually propagated to other hosts through forwarding *Location Update* packets. The host which initiates a location update is called an *issuer*. An issuer sends each *Location Update* packet to its peer (whose identifier shares a longest suffix with the issuer). An issuer

Table 1: The configuration parameters of an IEEE 802.11b wireless interface.

Application	Traffic	Constant bit-rate cbr
Transport	Control method	UDP
ROUTING	Location-aided routing protocol	
LL and ifq	Queuing model	CMUPriQueue
	Queue size	50 packets
MAC	IEEE 802.11b	Basic mode
PHY	Propagation model	TwoRayGround
	Data rate	2 Mbps
	Radius of signal coverage	250 meters
	Frame reception model	bit-error rate (BER)

host only unicasts a *Location Update* packet to a next-hop host in its current neighborhood. A *Location Update* packet can be further forwarded until no next-hop host can be found. The selection of a next-hop host follows Algorithm 1. Each host maintains a table for memorizing the locations of other hosts. Upon receiving a *Location Update* packet, a host updates its location table with the location information carried in the packet.

4.4. Answering Location Queries. A *target* host is the host whose location is under query. A *responder* host is the host which can answer the query about the location of a target host. A host looks up the location of a target host by sending a *Location Query* packet which includes the identifier of the target host and the last time when the location of the target host was updated. The last update time is used as the qualification criterion for selecting an up-to-date location from the copies of locations that are stored at different hosts. A host responds to the issuer of a *Location Query* by a *Location Response* packet which includes either the location of the target host, or a "not found" message when no answer can be found. A *Location Query* is failed to be answered when it cannot be further forwarded before an answer is found.

Upon receiving a *Location Query*, a host first looks into its location table for an answer. On a hit, this host responds to the issuer of this query with the copy found in its location table. On a miss, it forwards the query to a next-hop host for further service. The selection of a next-hop host makes a joint use of the peer-tree indexing structure and the underlying network connectivity. The selection of a next-hop host also follows the procedure illustrated in Algorithm 1.

5. Performance Analysis

The performance of the PTLS service has been evaluated by simulating wireless networks consisting of mobile hosts. The simulations are performed using the wireless and mobility extension [24] to the ns-2 simulator [25]. Each mobile host is equipped with an IEEE 802.11b wireless interface whose configuration parameters are shown in Table 1. Mobile hosts move within a 3000 × 3000 square meters, and their movements follow a random waypoint model [26]. The parameters adopted by the PTLS service is shown in Table 2.

5.1. Success Rates of Answering Location Queries. A location query is answered by either the up-to-date location upon a hit, or a "not found" message upon a query failure. The success rate of answering location queries is defined as the ratio of the number of location queries that are successfully answered to the total number of location queries that have been issued. The comparison of the success rates between the PTLS service and the GLS service [19] is shown in Figure 3(a). The success rate of the PTLS service is generally higher than GLS, especially when there are more than 100 hosts in a network. The higher success rates in PTLS signifies that forwarding chains are unlikely broken in the PTLS service when each mobile host is surrounded by many neighbors (ref. Figure 3(b)).

5.2. Bandwidth Consumptions. The comparison of the average per-host bandwidth consumption between PTLS and GLS is shown in Figure 4(a). The per-host bandwidth consumption goes up with increasing densities of mobile hosts in a network. PTLS consumes less amount of bandwidth than GLS. The number of times a service packet is forwarded highly contributes to bandwidth consumptions in a location service. Service packets in PTLS are forwarded almost only once on average as shown in Figure 4(c), while service packets in GLS generally take more hops as shown in Figure 4(d). This fact indicates that one service-level forwarding between two location servers in GLS may correspond to multiple geographic hops in the underlying network topology [19].

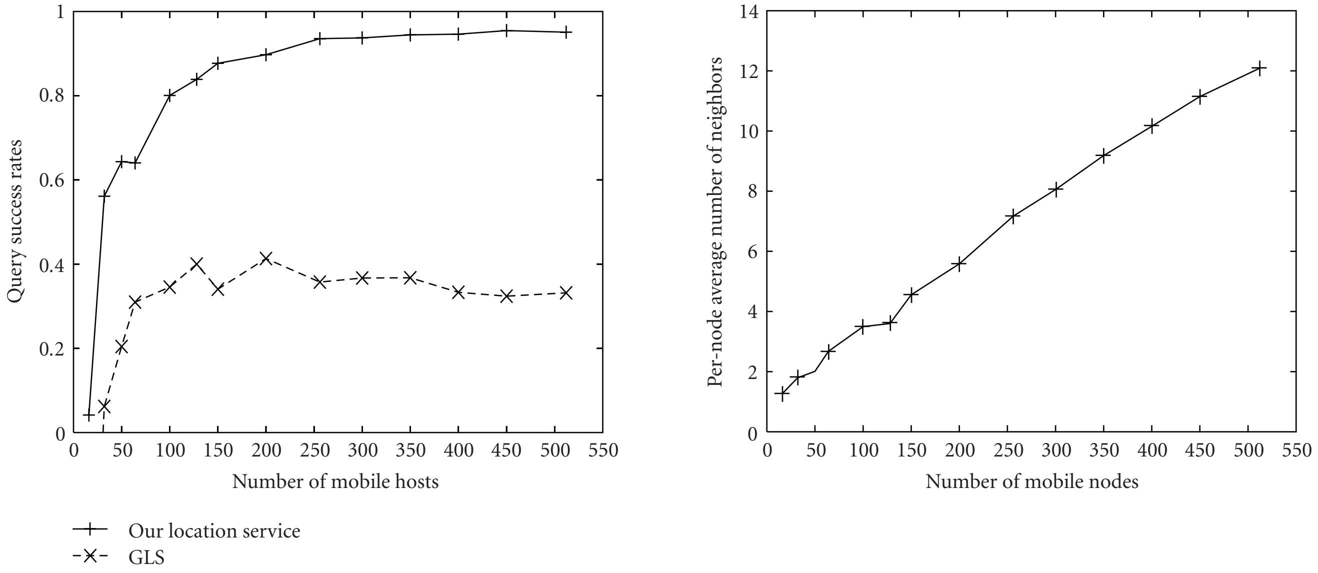
5.3. Robustness. The robustness to changes of network connectivity has also been evaluated in the existence of sleeping hosts and under various movement speed. The robustness of the PTLS service is shown in Figure 3(a).

(i) *Success rates in the existence of sleeping hosts*: Since the peer-tree is a complete binary search tree, it requires the number of leaf nodes to be an exact power of 2. When the number of hosts N is not an exact power of 2, a peer-tree is constructed with 2^l leaf nodes where $2^{l-1} < N < 2^l$. The number of missing leaf nodes in peer-tree is $2^l - N$. The missing leaf nodes can be imagined as the sleeping wireless hosts. The existence of sleeping hosts does not affect the success rates of answering location queries as shown in Figure 3(a). For example, the success rates under 50, 100, 150, 200, 250, 300, 350, 400, and 450 wireless hosts are comparable to the success rates under 2^6, 2^7, 2^8, 2^8, 2^8, 2^9, 2^9, 2^9, and 2^9 wireless hosts, respectively.

(ii) *Success Rates under Various Movement Speeds*: The performance of PTLS has also been validated by varying the movement speed of hosts. The results are very similar to the results shown in Figure 3. The robustness is mostly attributed to two facts. First, a location query can be answered by multiple hosts. Second, changes of network connectivity are unlikely to break the forwarding chains of location queries because a next-hop host is dynamically selected from multiple candidates where a copy of the up-to-date location can be found.

Table 2: The parameters of the PTLS service.

Hello broadcast	Broadcast interval	Broadcast once every 10 seconds of silence.
	Timeout	2∗ broadcast inverval
Location Update	Interval of update	Speed dependent Unicast a location update for every 100 meters of movement
	Timeout	2∗ current update inverval
Mobility	Model	Random waypoint model
	Range of mobility	A 2-D field of 3000 × 3000 square meters.
	Direction distance of movement	A host moves along a straight line to a randomly chosen destination.
	Pause time	Pauses for 4 seconds after arriving at a destination.
	Movement speeds	Between 5 and 6 meters per second.
	Duration	The movement lasts throughout a simulation.
Data Transmissions	Connections	Randomly chosen pairs of mobile hosts communicate to each other.
	Traffic model	Unidirectional (cbr) connections.
	Packet size	512 bytes of User Data + Headers
	Packet rate	10 packets per second
	Duration	Throughout the duration of a simulation which is 300 seconds.



(a) Success rates on location discovery

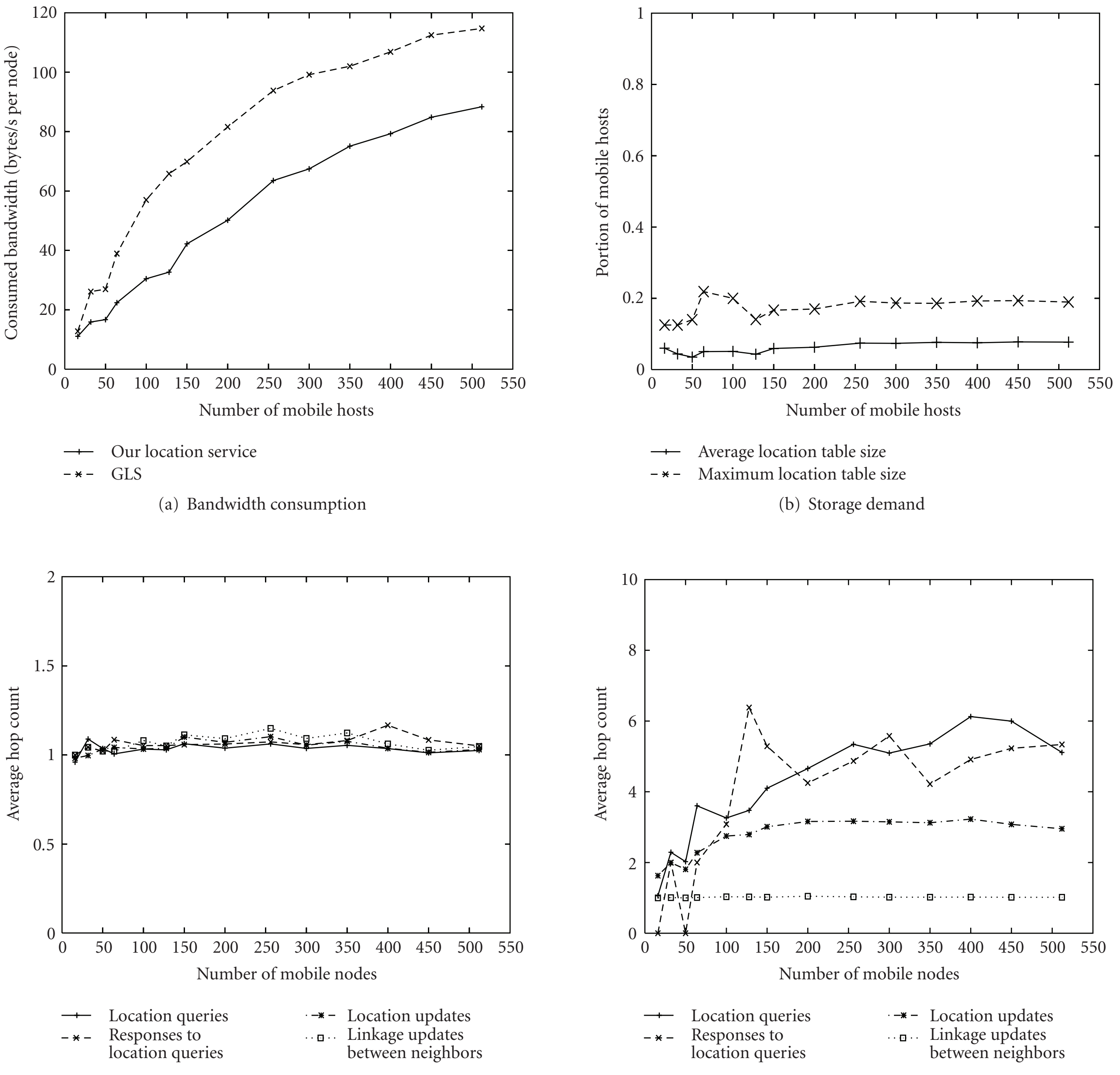
(b) Average number of neighbors

Figure 3: The performance of the PTLS service. Simulations are run in a 3000 × 3000 meter^2 square.

5.4. Storage Demand. The per-host storage demand for holding location replicas is evaluated by the ratio of the number of location replicas held in each host to the total number of hosts. The per-host storage demand in the PTLS service is shown in Figure 4(b). Each host only needs to hold location replicas for less than 10% of other hosts in most scenarios. The per-host storage demand is higher in some scenarios, but the storage demand is no more than 25%.

6. Conclusions

This paper has described a peer-tree based location lookup service. The goals of this location lookup service are to achieve high success rates on answering location queries, to keep low demands on network bandwidth and storage, and to be robust to changes of network connectivity. These goals are achieved by adopting a peer-tree indexing structure



(a) Bandwidth consumption

(b) Storage demand

(c) Average hop count in PTLS

(d) Average hop count in GLS

Figure 4: Comparison of the performance of the PTLS service and the GLS service under different numbers of mobile hosts. All simulations are run in a 3000 × 3000 meter^2 square.

which is built based on the relations of sharing various-length common suffixes of host identifiers. Guided by the indexing structure, each host periodically propagates its up-to-date location to a small number of other hosts. Similarly, location queries is served through the cooperation of a small set of hosts. Paths of forwarding the location updates or queries are jointly determined by the peer-tree indexing structure and the temporary network connectivity. Evaluated by simulating wireless networks with mobile hosts, the performance of the PTLS service exhibits high success rates on answering location queries, low consumption on network bandwidth, low demands on per-host storage, and robustness to changes in network connectivity.

References

[1] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 243–254, Boston, Mass, USA, August 2000.

[2] J. Liu, "Peer-tree: a peer-to-peer message forwarding structure for relaying messages in mobile applications," in *Proceedings of*

the 2nd International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P '05), pp. 87–94, La Jolla, Calif, USA, July 2005.

[3] W. Litwin, M.-A. Neimat, and D. A. Schneider, "LH: linear hashing for distributed files," in *Proceedings of the ACM International Conference on Management of Data (SIGMOD '93)*, pp. 327–336, Washington, DC, USA, May 1993.

[4] S. Mazuelas, A. Bahillo, R. M. Lorenzo, et al., "Robust indoor positioning provided by real-time RSSI values in unmodified wLAN networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 821–831, 2009.

[5] N. Patwari and A. O. Hero III, "Using proximity and quantized RSS for sensor localization in wireless networks," in *Proceedings of the 2nd ACM International Workshop on Wireless Sensor networks and Applications (WSNA '03)*, pp. 20–29, San Diego, Calif, USA, January 2003.

[6] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[7] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[8] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1734–1743, San Francisco, Calif, USA, March-April 2003.

[9] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, vol. 6, no. 2, pp. 46–55, 1999.

[10] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the ACM International Conference on Communications Architecture and Protocols (SIGCOMM '94)*, pp. 234–244, London, UK, August-Semptember 1994.

[11] P. F. Tsuchiya., "The landmark hierarchy: a new hierarchy for routing in very large networks," in *Proceedings of the ACM International Conference on Communications Architecture and Protocols (SIGCOMM '88)*, pp. 35–42, Stanford, Calif, USA, August 1988.

[12] M. Gerla, X. Hong, and G. Pei, "Landmark routing for large ad hoc wireless networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '00)*, vol. 3, pp. 1702–1706, San Francisco, Calif, USA, November-December 2000.

[13] Y.-B. Ko and N. H. Vaidya., "Location-aided routing (LAR) in mobile ad-hoc networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom '98)*, pp. 66–75, Dallas, Tex, USA, October 1998.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," in *Proceedings of the ACM International Conference on Communications Architecture and Protocols (SIGCOMM '01)*, pp. 161–172, San Diego, Calif, USA, August 2001.

[15] P. Bozanis and Y. Manolopoulos, "LDT: a logarithmic distributed search tree," in *Proceedings of the Workshop on Distributed Data and Structures (WDAS '02)*, Paris, France, March 2002.

[16] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup service for internet applications," in *Proceedings of the ACM International Conference on Communications Architecture and Protocols (SIGCOMM '01)*, pp. 149–160, San Diego, Calif, USA, August 2001.

[17] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: a resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, 2004.

[18] K. Aberer, "P-Grid: a self-organizing access structure for P2P information systems," in *Proceedings of the International Conference on Cooperative Information Systems (CoopIS '01)*, Trento, Italy, September 2001.

[19] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 120–130, Boston, Mass, USA, August 2000.

[20] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *Proceeding of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1190–1199, New York, NY, USA, June 2002.

[21] D. Johnson, D. Maltz, and J. Broch, "DSR: the dynamic source routing protocol for multihop wireless ad hoc networks," in *Ad Hoc Networking*, C. E. Perkins, Ed., pp. 139–172, Addison-Wesley, Reading, Mass, USA, 2001.

[22] C. Maihofer, "A survey of geocast routing protocols," *IEEE Communications Surveys and Tutorials*, vol. 6, no. 3, pp. 32–42, 2004.

[23] K. Kar, M. Kodialam, and T. V. Lakshman, "Routing restorable bandwidth guaranteed connections using maximum 2-route flows," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 772–781, 2003.

[24] "The wireless and mobility extensions to NS-2," Monarch Project, Computer Science Department, Rice University, http://www.monarch.cs.rice.edu.

[25] "The Network Simulator—ns-2," UC Berkeley, LBL, USC/ISI, and Xerox PARC, http://www.isi.edu/nsnam/ns.

[26] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom '98)*, pp. 85–97, Dallas, Tex, USA, October 1998.