Location Information Services in Mobile Ad Hoc Networks[*]

Tracy Camp
Department of Math. and Computer Sciences
Colorado School of Mines
Golden, Colorado 80401
tcamp@mines.edu

October 21, 2003

Abstract

In recent years, several position-based routing protocols have been developed for mobile ad hoc networks. Many of these protocols assume a location service is available that provides location information on the nodes in the network. In this chapter, we survey all the proposed location information services that exist in the literature to date. We classify these location information services into three categories: proactive location database systems, proactive location dissemination systems, and reactive location systems.

1 Introduction

Routing a packet from a source to a destination in a mobile ad hoc network is a challenging problem, since nodes in the network may move and cause frequent, unpredictable topological changes. Several unicast routing protocols have been proposed for MANETs in an attempt to solve the routing problem. Example protocols include Dynamic Source Routing (DSR) [38, 39], Ad hoc On demand Distance Vector (AODV) [56, 58], and the Zone Routing Protocol (ZRP) [26]; a survey of these protocols is available in [60] and [66] and a performance comparison for a few of the protocols are in [10], [37], [18, 57], and [17].

In an effort to improve the performance of unicast communication, some of the proposed MANET unicast routing protocols use location information in the routing protocol. A few of the proposed algorithms include the Location-Aided Routing (LAR) algorithm [44], the Distance Routing Effect Algorithm for Mobility (DREAM) [4], the Greedy Perimeter Stateless Routing (GPSR) algorithm [40], the Geographical Routing Algorithm (GRA) [35], the Terminode Remote

---

[*] This work supported in part by NSF Grants ANI-0073699 and ANI-0240588. Research group's URL is *http://toilers.mines.edu*. Reference for this manuscript: Technical Report MCS-03-15, The Colorado School of Mines, October 2003.

Routing (TRR) protocol [5, 6], the Scalable Location Update-based Routing Protocol (SLURP) [68], the Depth First Search (DFS) algorithm [64], Greedy-Face-Greedy (GFG) algorithm [19], and the GPS Zone Routing Protocol (GZRP) [9]. Surveys of position-based routing protocols for ad hoc networks are available in [23, 24, 51, 63].

Several benefits of position-based routing protocols exist:

1. Reduced overhead, since the establishment and maintenance of routes is (usually) not required in a protocol that uses location information for routing. In position-based routing protocols, each node maintains accurate neighborhood information and a rough idea of the destination's location.
2. Scalability, since routing protocols that do not use location information do not scale [35, 46].
3. Localized algorithm, since a node determines which neighbor to forward a message only on its neighborhood and the destination's location.
4. Higher performance, as shown in the simulation results of several articles (e.g., [13] and [44]).

One of the main challenges of a position-based routing protocol is to learn an accurate position or location for a packet's destination. While some protocols (e.g., DREAM and LAR) include the exchange of location information as a part of its protocol, most position-based routing protocols assume a separate mechanism that provides location information. For example, knowledge about the location of a destination node is assumed available in all geographic forwarding routing protocols (e.g., [20, 40, 65]). In fact, in the simulation results presented for these protocols, location information is typically provided to all mobile nodes without cost [20, 40]. Thus, overhead in simulation results on protocols such as DREAM are much higher, because DREAM includes the task of maintaining location information on destination nodes. This paper is a survey of protocols that provide location information in an ad hoc network.

As mentioned, in a position-based routing protocol, each node needs to determine its location in the ad hoc network. A node can determine its location if the Global Positioning System (GPS) is available. If GPS is unavailable, a relative coordinate system, such as the ones proposed in [5, 11, 15, 16], can be used. In a relative coordinate system, a node estimates its distance from a neighbor using signal strength. Absolute and relative location systems for indoor and outdoor environments are surveyed in [30]. Finally, if some nodes have the ability to determine their locations, [20] proposes that location aware nodes serve as location proxies for location ignorant nodes.

The rest of this paper is organized as follow. First, in Section 1.1, we detail one popular type of position-based routing algorithm (i.e., geographic forwarding). As mentioned, geographic forwarding algorithms require a separate mechanism (called a location service) to provide location information on nodes in the ad hoc network. In Section 1.2, we give an overview of the location services surveyed and discuss how we classify the surveyed location services in the rest of this paper. The next three main sections (i.e., Sections 2–4) detail the location services surveyed. Finally, in Section 5, we state our conclusions and comment upon the future work that is needed on location services.

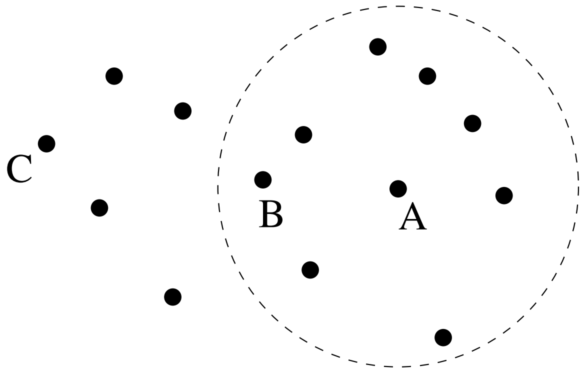

Figure 1: An example of the Most Forward with fixed Radius (MFR) scheme.

1.1 Geographic Forwarding

When location information is available, geographic forwarding can be used in place of establishing a route from a source node to a destination node. In geographic forwarding, which is a greedy scheme, each node forwards the packet to the neighbor closest to the packet's destination [21, 31, 65]; in other words, the packet is sent to a neighboring node that makes the most forward progress. (HELLO packets are used to maintain neighborhood information.) The goal of this strategy, which is called *Most Forward with fixed Radius (MFR)* in [31], is to minimize the number of hops to the destination. Figure 1 shows an example of this greedy MFR scheme. If node *A* is forwarding a packet toward *C*'s location, node *A* chooses node *B* to forward the packet further.

We note that [31] evaluates three strategies in choosing a node to forward a packet, and two of these strategies use variable transmission power. The authors determined that a network can obtain higher throughput if a node adjusts its transmission power to reach the nearest node in a forward direction. In other words, fewer conflicts occur in this strategy than in the one that minimizes the number of hops to the destination.

If no neighbor node is closer to the packet's destination than the node currently holding the packet, then the packet has reached a "dead-end"[1]. Several solutions have been proposed as a recovery strategy to a dead-end situation (e.g., [7, 8, 19, 20, 21, 35, 40, 65, 68]). In [21], a node recursively searches its neighbor's neighbors for a node that is closer to the destination than itself. If no node exists in a forward direction, [65] proposes the packet is sent to the closest node in a backward direction. A node asks its neighbors for an alternative route to the destination in Scalable Location Update-based Routing Protocol (SLURP) [68]. In the Greedy Perimeter Stateless Routing (GPSR) [40] protocol, a planar subgraph is created to route around the dead-end; this method was first proposed in [7, 8, 19]. If a packet does not reach its destination via geographic forwarding, [20] proposes the source node randomly choose an intermediate node through which to forward packets; incorporating a stop at an intermediate node on the way to the destination may allow the packet to bypass a dead-end situation. Finally, a route discovery is initiated in the Geographical Routing Algorithm (GRA) for a "dead-end" situation [35].

---

[1] If the network is dense, a "dead-end" situation should not occur [32, 46].

The authors of [68] mention that precise coordinates on the destination are unknown when the first data packet is transmitted; instead, the source only knows the home region (see Section 2.1) where the destination is located. Thus, SLURP routes packets with MFR toward the center of the home region. Once a node inside the home region receives the packet, SLURP uses DSR [38, 39] to determine a route to the destination within the home region.

1.2 Location Services

As mentioned, a location service is responsible for providing location information on nodes in the network. There are two general types of location services: proactive location services and reactive location services. Proactive location services are those protocols that have nodes exchange location information periodically. Reactive location services query location information on an as needed basis.

Proactive location services can be further classified into two general types: location database systems and location dissemination systems [32]. In a location database system, specific nodes in the network serve as location databases for other specific nodes in the network. When a node moves to a new location, the node updates its location database servers with its new location; when location information for a node are needed, the node's location database servers are queried. In a location dissemination system, all nodes in the network periodically receive updates on a given node's location. Thus, when a given node requires location information on another node, the information is found in the node's location table. "A location dissemination system can be treated as a special case of a location database system, where every node in the network is a location database server for all the others" [32].

In comparing location database systems to location dissemination systems, we find that:

- A node maintains more state in a location dissemination system than in a location database system, since location information is maintained on all nodes in the network.
- An update in a location dissemination system is (usually) more costly than an update in a location database system, since an update is disseminated to all nodes in the network.
- A location dissemination system is more robust than a location database system, since location information is maintained by all nodes in the network.
- A location query in a location dissemination system is (usually) more cheaper than a location query in a location database system, since the desired location information is available in a node's location table.

In [51], location services are classified by:

1. how many nodes host the service (i.e., some specific nodes or all nodes) and
2. how many positions of nodes a location server maintains (i.e., again, some specific nodes or all nodes).

As an example, in a traditional cellular network several dedicated location servers maintain location information about all nodes. Thus, a cellular network would be classified as a *some-for-all* approach. Typically, location database systems would be classified as an *all-for-some* approach and location dissemination systems would be classified as an *all-for-all* approach. In our discussion of location services, we refer to this classification as the *Mauve Classification*.

To maintain location information on other nodes in the network, we assume that each mobile node maintains a location table. This table contains an entry on every node in the network whose location information is known, including the node's own location information. A table entry contains (at least) the node identification, the coordinates of the node's location based on some reference system, and the time this location information was obtained. When a location request occurs, a node will first look in its location table for the information. If the information is not available in the table, a location database system will initiate a location query; if no result is returned, the node periodically transmits queries according to a timeout interval. On the other hand, a location dissemination system will (typically) flood a location request packet to all nodes in the MANET. A reply to this location request packet is transmitted by the node whose location was requested; nodes that receive a reply to a location request update their tables in a promiscuous manner.

We survey proactive location database systems in Section 2. In Section 3 we survey proactive location dissemination systems. Lastly, in Section 4, we survey reactive location systems.

2 Proactive Location Database Systems

In this section, we survey two types of proactive location database systems: home region location services and quorum-based location services. In each type, we survey four protocols that have been proposed in the literature. As mentioned, in a location database system, specific nodes in the network serve as location databases for other specific nodes in the network. A node will update its location database servers when it moves to a new location; a node's location database servers are then queried when location information on the node is needed. According to the *Mauve Classification* [51], most location database systems would be classified as an *all-for-some* approach. That is, all nodes in the network serve as database servers, and each database server maintains location information for some specific nodes in the network.

2.1 Home Region Location Services

Several home region location services, which are similar to Mobile IP [55] and our cellular phone network, have been proposed [22, 62, 68]. In these protocols, each node in the network is associated with a home region. A home region (called virtual home region in [5, 22, 34] and home agent in [62]) is either defined by a rectangle or a location and a circle, with radius R, around that location. In [62], the center of the circular home region for a node is defined by the node's initial location in the network; each node informs every other node about its initial location via a network-wide broadcast. In [5, 22, 34], the center of the circular home region is derived by apply-

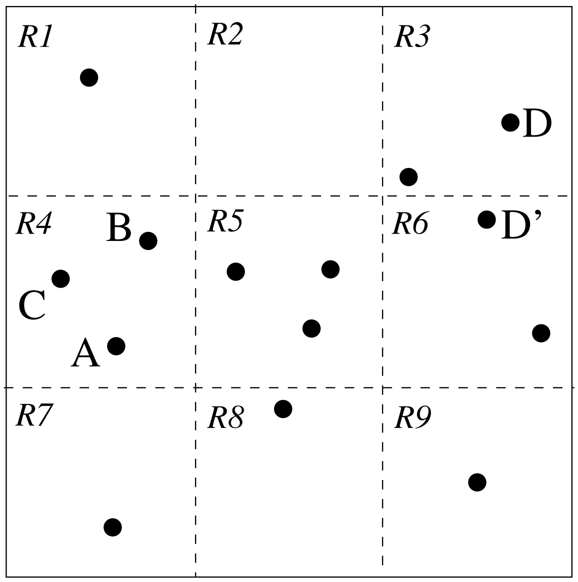

Figure 2: An example of the Home Region Location Service.

ing a well-known hash function on the node's identifier. The ad hoc network in [68] is logically arranged into equal-sized rectangular regions, and a static mapping, f, associates a node's identifier (ID) with a specific region. This many-to-one mapping is known by all nodes in the network [68].

If the number of nodes in the home region is too high, which leads to too much overhead, or too low, which leads to a lack of robustness, the radius of the home region is modified in [5, 22, 34]. In other words, the radius R dynamically adapts with a goal of maintaining an approximate constant number of nodes in the home agent.

All nodes within a given node's home region (i.e., all nodes currently within the defined circle or rectangle) maintain location information for the node. Thus, when a node moves to a new location, it sends its current location to the nodes in its home region. In [62], a location update is transmitted when the number of created or broken edge links reaches a fixed threshold. In [5], three location update schemes are proposed: timer-based (i.e., periodic updates), distance-based (i.e., update when the node has moved more than a threshold), and predictive distance-based. In the predictive distance-based scheme, a node sends its location and velocity to the home agent. The node transmits a location update when the difference between the predicted location and the node's actual location is greater than a threshold. Lastly, in [68], a node transmits a location update when it moves from one home region to another.

Figure 2 gives an example of the rectangular home region approach. Suppose the home region of node D is $R4$. When D moves from $R3$ to $R6$, nodes A, B, and C will receive the location update packet.

A location update packet travels to a home region via a geographical forwarding protocol. When a node inside the home region receives a location update packet, the node broadcasts the packet to all nodes within the home region. To decrease the overhead in this broadcast, [68] suggests that a node include its neighbor list in each location update packet. When a node inside the home region receives a location update packet, it re-broadcasts the packet only if new neighbors will be reached. This technique is called Flooding with Self Pruning [49, 67].

When a source wants to transmit a packet to a destination, the source queries the destination's home agent for the destination's current location. Specifically, a location request packet is sent from the source to the center of the home region via a geographical forwarding protocol. Each node that receives this request inside the home region responds with the current location of the destination [5]. If a location cache is recent enough (i.e., 10 seconds or less), an intermediate node responds to the location request in [68]. To avoid a reply storm, the location response is transmitted after a random waiting period based on the age of the cached entry.

The authors of [68] give several details for their home region location service approach. For example, they propose a solution for when no nodes are currently in a given node's home region (e.g., $R2$ in Figure 2). In this situation, all the regions surrounding the empty home region become home regions for nodes that map to the empty home region. Thus, if a home region is in the center of the network, the eight surrounding rectangular regions become proxies for the empty home region. For example, in Figure 2, regions $R1$, $R3$, $R4$, $R5$, and $R6$ all act as a home region for nodes whose identifiers map to $R2$. See [68] for full details on the maintenance of home regions.

A fourth home agent approach, the Hierarchical State Routing (HSR) protocol, has also been proposed [53, 54]. The main difference between this scheme and the three previously discussed is that the HSR protocol assumes the network is organized hierarchically. Thus a location update/request must travel along a hierarchical tree to the home agent.

2.2 Quorum-Based Location Services

Several quorum-based location services have been developed and are based on replicating location information at multiple nodes that act as information repositories [2, 27, 28, 41, 46, 48, 61]. In other words, read and write quorums (i.e., subsets of the nodes) are defined in the network. When a node determines a location update is needed, it transmits the update to a write quorum. When a source node wants to transmit a packet to a destination node, it requests the location for the destination from a read quorum. The main challenge in a quorum-based system is to define the read/write quorums in such a way as to maximize the probability of a query success. That is, the goal is to define the quorums such that a read quorum for a node intersects the write quorum for any other node. Thus, up-to-date location information can be obtained for any given destination.

One simple solution to forming the read/write quorums is as follows: the write quorum for any node i is i and the read quorum for any node j is all nodes in the network. In this solution, the write quorum is of size 1 and the read quorum is of size n where n is the number of nodes in the network. Thus, while the cost of a location update is small, the cost of a location request is very large. In this section, we survey quorum-based location services that attempt to minimize the combined cost of updates and queries.

2.2.1 Uniform Quorum System (UQS)

The Uniform Quorum System (UQS) is presented in [27, 28, 48]. During the initial setup of UQS, a subset of the network nodes are chosen that best serve as the network's virtual backbone (VB); for example, chosen nodes may be those that are uniformly distributed throughout the network. (See [48] for implementation issues concerning the VB, e.g., the maintenance of the VB in the presence of node disconnections.) Quorums are then defined as subsets of the VB nodes, such that any two quorums intersect.

To update its location, a node transmits its new location to the nearest VB node. The VB node then sends the location update to a randomly chosen quorum. In other words, there is no fixed association between the node and the quorum updated. The procedure for a location request is similar: a node transmits a location request to its nearest VB node, which then contacts a randomly chosen quorum for the location.

Haas and Liang propose a node updates its location in three different manners:

1. VB change: whenever the node's nearest VB node changes, a location update occurs.
2. Location request: whenever the node requests a location, a location update occurs in the queried quorum.
3. Periodically: whenever a given time period has elapsed without an update, a location update occurs. This update will ensure timestamps on location information are kept current.

Haas and Liang discuss several methods to generate quorums in [28]. Since the size of the quorum intersection is a configuration parameter, an implementor of UQS can decide the overhead to resiliency tradeoff. That is, a larger quorum size means a larger cost for location updates and location requests; however, a larger quorum size also means a larger quorum intersection. Regarding *Mauve's Classification*, UQS can be configured to operate as an all-for-all, all-for-some, or some-for-some approach. Since the VB nodes are typically a subset of all network nodes and the nodes in a quorum are typically a subset of the VB nodes, UQS will most often be configured as a some-for-some approach.

2.2.2 A Column/Row Method

Another quorum-based location service is proposed in [61]. When a node determines a location update is needed[2], the node transmits its new location to all nodes located in its *column*. A node's *column* is all the nodes to the north and south of the node's current location. In other words, the node transmits a location update packet that travels in both the north and south direction. While the thickness of the column can be configured, [61] defines a column as one path for ease of understanding. All nodes on the path, and all neighbors of nodes on the path, receive the updated location information. To increase the column size to two, neighbor nodes would need to re-transmit each location update.

---

[2] Due to results in [41], [61] chose to transmit a location update when the number of created or broken edge links reaches a fixed threshold. See Section 2.2.5.

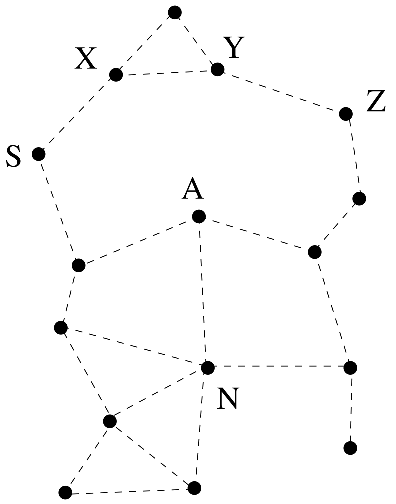

Figure 3: A problem with the Column/Row Location Service.

When a source node initiates a location query for a destination, the location request is first sent to all neighbors within q hops of the source. If there is no reply to this request, the location request proceeds in the same manner as a location update. That is, a source node transmits a location request in its *row*, which traverses in an east-west direction. (Again, the thickness of the row can be configured.) The location request packet contains the time of the most recent location information known by the source. If a node that receives the location request has more recent location information, a location reply packet is sent. Once the most easternmost/westernmost node is reached, the location request packet is transmitted via a geographic forwarding protocol toward the most recent location information on the destination. A second location request packet can be sent from the source node as well. This location request is transmitted toward the most recent location information on the destination using a geographic forwarding protocol.

One problem with this column/row method is illustrated in Figure 3. When node N transmits a location update, its most northernmost node is currently A. If the location update terminates at A, then a search by source S, which follows the $X-Y-Z$ path, will not return the most recent location information on node N. To ensure an intersection between the north/south and east/west directions occurs, [61] proposes that node A switch to FACE mode (see [7, 8]) until another node, more northern, is found.

2.2.3 Grid Location Service (GLS)

Another quorum-based location database system in an ad hoc network is called the Grid Location Service (GLS) [46]. The set of GLS location servers is determined by a predefined geographic grid and a predefined ordering of mobile node identifiers in the ad hoc network. There are three main activities in GLS, which we detail in the following discussion: location server selection, location query request, and location server update.

Initially, the area covered by the ad hoc network is arranged into a hierarchy of grids with

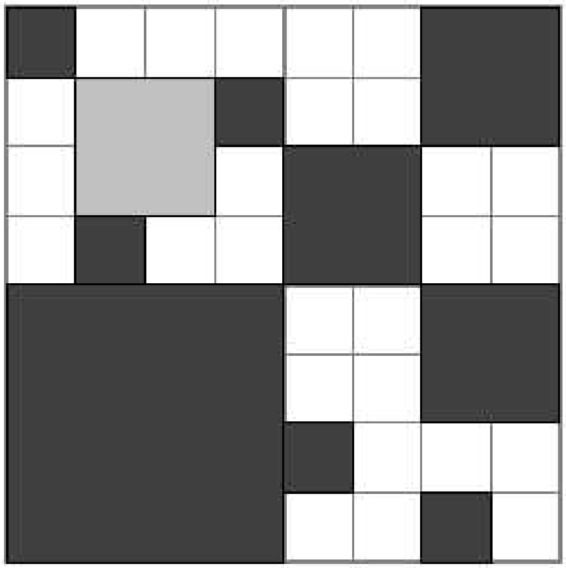

Figure 4: An example grid.

squares of increasing size. The smallest square is called an order-1 square. Four order-1 squares make up an order-2 square, four order-2 squares make up an order-3 square, and so on. A few example squares of various orders are shown in Figure 4 with dark shading. (This figure is taken from [46].) Specifically, five order-1 squares, three order-2 squares, one order-3 square, and one order-4 square are shown. An order-n square's lower left coordinates must be of the form (a2^{n-1}, b2^{n-1}) for integers a, b [46]. Thus, in Figure 4, the lower left coordinates of the lightly shaded square is (1, 5); although this is an example of a 2x2 square, it is not an order-2 square since no integers a and b exist such that (2a, 2b) = (1, 5).

Location Server Selection: A node chooses its location servers by selecting a set of nodes with IDs close to its own ID. Each of the chosen location servers have the least ID greater than the node's ID in that order square. Figure 5, which is taken from [46], provides an example of how node B selects its location servers. (B's location servers are circled in the figure.) In this example, B (ID 17) determines which nodes will be its location servers by selecting nodes with IDs closest to its own. As mentioned, a node is defined as closest to B when it has the least ID greater than B. In other words, the location server's ID is the smallest number that is greater than B's ID. For example, consider the grid to the left of B's grid. No ID exists that is greater than 17. Since the ID space is considered to be circular (i.e., wraps around), 2 is defined as closer to 17 than 7. B selects three location servers for each level of grid order square, which combine to make the next level of grid order square. For example, in Figure 5, B selects one location server from each order-1 square (e.g., 2, 23, and 63) that, along with its own order-1 square, will make an order-2 square. B then selects one location server from each order-2 square (e.g., 26, 31, and 43) that, along with its own order-2 square, will make an order-3 square. Each of the chosen location servers have the least ID greater than B in that order square.

Location Query Request: As a quorum-based location service, a node initiates a location

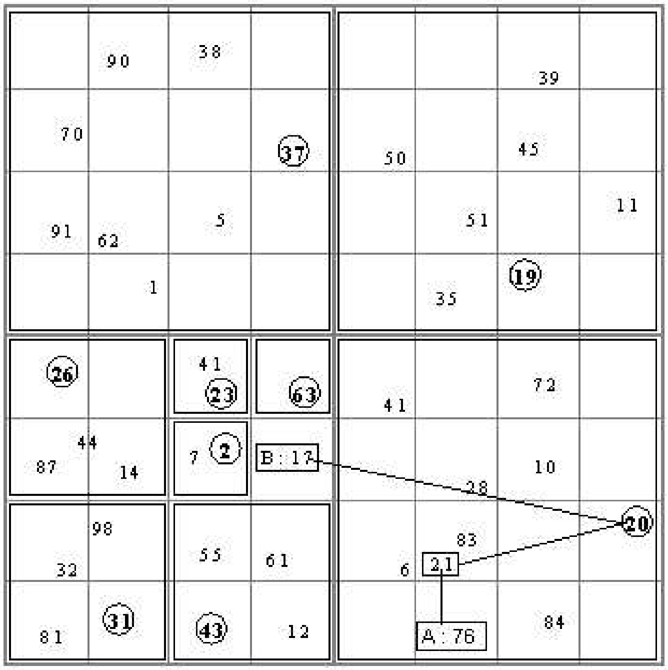

Figure 5: A GLS example.

query request when the node needs a location for a destination. Since each node knows all nodes within its order-1 square, the request is first sent to a potential location server for the destination desired in the requesting node's order-2 square. In other words, the location query request packet is forwarded, using geographic forwarding, to a node whose ID is the least greater than or equal to the ID of the destination within the order-2 square. That node then forwards the query using the same algorithm until it reaches a location server for the destination. This location server then forwards the query directly to the destination, which responds to the location query request with its most recent location.

Figure 5 illustrates the path of a query packet from A to B. A (ID 76) sends a location query to node with ID 21, since A knows the location of node 21 and it is the least greater than B (ID 17) in that grid order square. In other words, no node with IDs between 17 and 21 exist in A's order-2 square. Node 21 then forwards the query packet using the same algorithm to the node in the next order square in the grid hierarchy, which is ID 20 in this example. Since node 20 is a location server for B in that grid order square, it knows B's location and is able to forward the query packet directly to B. Since the query packet contains A's location, B responds by sending its current location to A using geographic forwarding.

Location Server Update: Since GLS is a quorum-based location service, a location server update occurs when a node moves. Each node, acting as a location server for the nodes it serves, maintains two tables. The location table holds the location of nodes that have selected this node as its location server; each entry contains a node's ID and geographic location. A node also maintains a location cache, which is used when a node originates a data packet; the cache holds information

HELLO		
Source ID	Source location	Source spe ed
Neighbor list: IDs and location		
Forwarding pointers		

Table 1: HELLO packet fields

from the update packets a node has forwarded. Because a node uses the routing table maintained by geographic forwarding for its order-1 square neighbors, it does not need to send GLS updates within its order-1 square.

When a node moves a given threshold, it must send an update packet to all of its location servers. To avoid excessive update traffic, the update frequency is calculated using a threshold distance and the location servers' square order [46]. The threshold distance is the distance the node has traveled since the last update. For example, suppose a node updates its location servers in order-2 squares when it moves a distance d; the node then updates its location servers in order-3 squares when it moves a distance 2d. In other words, a node updates its location servers at a rate proportional to its speed, and the distant location servers are updated less frequently than the nearby location servers.

Before a node sends a data packet, it checks its location cache and location table to find the location of the destination. If it finds an entry for the destination, it forwards the data packet to that location. Otherwise, it initiates a location query using GLS, and stores the data packet in a buffer waiting for the query result. If no result is returned, the node periodically transmits queries according to a timeout interval. Once it gets the query result, it will use geographic forwarding to send the data packet.

Location Query Failures: There are two types of failure caused by node mobility [46]: a location server may have out-of-date information or a node may move out of its current grid. The solution for the first type of failure is to use the old location information. To overcome the second type of failure, which is more common, the moving node places a forwarding pointer in the departed grid. This forwarding pointer points to the grid the node has just entered. In other words, before a node moves out of a grid, it broadcasts its forwarding pointer to all nodes in the grid. Any node in this grid stores the forwarding pointer associated with the node that just left the grid; a node discards forwarding pointers when it departs a grid.

To share forwarding pointers with other nodes that have entered the order-1 square, a randomly chosen subset of forwarding pointers are transmitted with each HELLO packet [46]. (Table 1 shows the HELLO packet fields.) A node receiving a HELLO packet adds forwarding pointers to its own collection of forwarding pointers only if the broadcaster of the HELLO packet is in the same grid as the node's grid. A forwarding pointer allows a data packet to be forwarded to a grid that may contain the node. We note that a dense network is needed for the forwarding pointers to be effective and all simulation results presented in [46] use dense networks. See [25], [36] and [43] for further performance results on GLS.

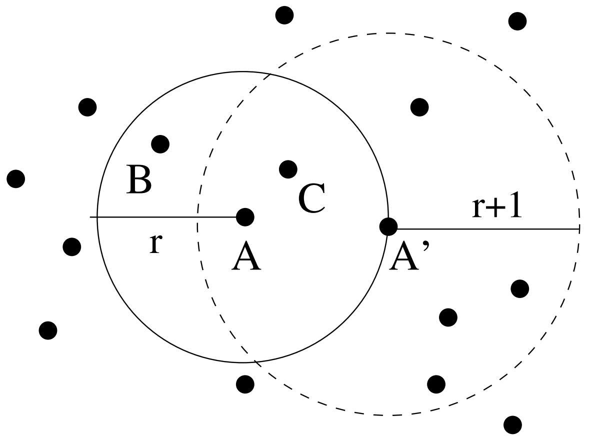

Figure 6: Example of location update in the Doubling Circles scheme.

2.2.4 Doubling Circles

The location service in [2] is similar to GLS [46]. However, instead of arranging the network into squares of increasing size, [2] arranges the network into circles of increasing size centered on a node's location; the radius of each subsequent circle is one meter larger than the radius of the previous circle.

Suppose a node was at location *i* when it transmitted its last location update. When the node moves outside a circle of radius *r* meters centered at location *i*, the node transmits a location update to all nodes within a circle of radius $(r+1)$ meters centered at the node's new location. Figure 6 illustrates an example. When node *A* moves to A', it transmits a location update packet to all nodes within the dashed circle of radius $(r+1)$. Multipoint Relaying [59, 67], an efficient network-wide broadcast protocol, is used to limit the overhead associated with flooding a location update packet within a given circle.

Similar to other location services, the distance a location update packet propagates corresponds to the distance the node has moved since its previous update [2]. In other words, nodes close to a given destination will have more accurate location information on the destination than nodes farther away from the destination. Thus, when a packet is transmitted toward a destination, forwarding nodes re-direct the packet with more up-to-date location information on the destination. For example, node *B* in Figure 6 forwards a packet for *A* to the center of the solid circle. When node *C* receives this packet, it re-directs the packet to the center of the dashed circle.

2.2.5 Quorum-Based Details

As mentioned, the main challenge in a quorum-based system is to define the read/write quorums in such a way as to maximize the probability of a query success. This challenge is especially difficult in a mobile ad hoc network, since network partitions are possible. The authors of [41] propose a quorum-based location service for information dissemination in mobile ad hoc networks with partitions. In [41], the authors assume that the quorum sets are constructed a priori, that at least one node intersects any two quorum sets, and the quorum sets are known by every node in the network. Although the article does not propose a scheme to configure the quorums, it does attempt to answer three important questions:

1. When should a node transmit a location update?
2. Where should a node transmit a location update?
3. Which nodes should a node query for location information?

In [41], four strategies are proposed to answer *When should a node transmit a location update?* Based on their performance evaluation, the "absolute connectivity-based policy" was found to be the best trigger update strategy. In this strategy, a node sends an update "when a certain *pre-specified number* of links incident on it have been established or broken since the last update" [41]. The authors of [41] argue that absolute distance-based updates and velocity-based updates are ineffective.

Three strategies are proposed to answer the remaining two questions: *Where should a node transmit a location update?* and *Which nodes should a node query for location information?* In all three strategies, a node uses reachability knowledge to determine which quorum set to choose for a location update or location request. That is, a node uses knowledge of unavailable nodes to select a quorum set in the hopes of minimizing a current or future location request failure. In their performance evaluation, the authors found the Eliminate_then_Select (ETS) Strategy was the best location update strategy and the Select_then_Eliminate (STE) Strategy was the best location request strategy in choosing a quorum set to update/query. In ETS, a node first eliminates all quorum sets that have at least one unreachable node. A quorum set is then randomly chosen from the remaining sets. The ETS strategy maximizes the number of nodes that receive an update. In STE, a node first randomly chooses a quorum set and then eliminates unreachable nodes from the set. The STE strategy maximizes the number of quorum sets to query.

3 Proactive Location Dissemination Systems

In this section, we survey six proactive location dissemination systems. As mentioned, all nodes in the network periodically receive updates on a given node's location in a location dissemination system. Thus, location information should be available for every node in a given node's location table. According to the *Mauve Classification* [51], location dissemination systems would be classified as an *all-for-all* approach. That is, every node in the network maintains location information on every other node in the network.

3.1 DREAM Location Service (DLS)

The DREAM Location Service (DLS) [12] is similar to a location service proposed by the authors of DREAM [4], a position-based routing protocol. Each location packet (LP), which updates location tables, contains the coordinates of the source node based on some reference system, the

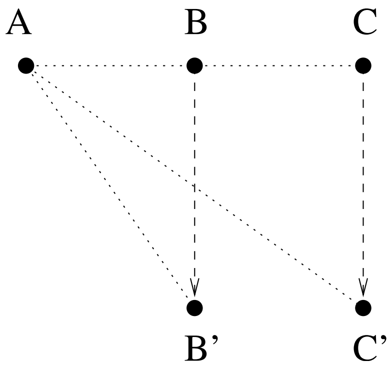

Figure 7: The distance effect: although B and C have moved the same distance, from A's perspective, $\widehat{BAB'} > \widehat{CAC'}$.

source node's speed, and the time the LP was transmitted. Each mobile node in the ad hoc network transmits an LP to nearby nodes at a given rate and to faraway nodes at another lower rate. The rate a mobile node transmits LPs adapts according to when the mobile node has moved a specified distance from its last update location. Since faraway nodes *appear* to move more slowly than nearby mobile nodes, it is not necessary for a mobile node to maintain up-to-date location information on faraway nodes. This phenomenon is termed the *distance effect* and is illustrated in Figure 7. (This figure is taken from [3].) By differentiating between nearby and faraway nodes, the overhead of location packets is limited.

DREAM controls the accuracy of the location information in the network via the frequency that it transmits LPs and how far the LP is allowed to travel [4]. However, details on implementing the proposed location service are lacking in [4]. The DLS solution for the transmission of LPs in [12] is:

transmit nearby LP: $(\frac{T_{range}}{\alpha}) * (\frac{1}{\nu}) = \frac{T_{range}}{\alpha\nu}$
transmit faraway LP: one for every X nearby LPs
or
at least every Y seconds,

where T_{range} is the transmission range of the mobile node, ν is the average velocity of the mobile node, and α is a scaling factor. In DLS, LPs to faraway nodes update all nodes in the network, including nearby nodes.

There are some similarities of DLS to the Internet standard Open Shortest Path First (OSPF) [52]. Specifically, both transmit information concerning the local environment to all nodes in the network on a periodic basis. There are, however, three significant differences between these two protocols. First, DLS differentiates between nearby nodes and faraway nodes; OSPF has no such differentiation. Second, a node using DLS will transmit its location information; a node using OSPF will transmit its neighbor information. Lastly, OSPF was not developed for an ad hoc network environment; thus, the rate of OSPF update packets is only based on time not on distance moved.

3.2 Simple Location Service (SLS)

The Simple Location Service (SLS) is also a proactive dissemination location service, except this service only transmits location information to neighbors [12]. Specifically, each location packet (LP), that updates location tables, contains the location of several nodes, the speed of each of these nodes, and the time the LP was transmitted. The rate a mobile node transmits LPs adapts according to location change:

$$\left(\frac{T_{range}}{\alpha}\right) * \left(\frac{1}{\nu}\right) = \frac{T_{range}}{\alpha\nu}$$

or

at least every Z seconds,

where T_{range} is the transmission range of the node, ν is the average velocity of the node, and α, which is a constant optimized through simulation, is a scaling factor. For a given T_{range}, the frequency a mobile node transmits LPs adapts according to the time used by the mobile node to move a specified distance from its last update location. The faster the node moves, the higher frequency the mobile node transmits LPs.

In SLS, each LP contains up to E entries from the node's location table [12]. These E entries are chosen from the table in a round robin fashion. That is, each LP transmission shares location information on several nodes in the MANET with the node's neighbors. As multiple LPs are transmitted, all the location information a node knows is shared with its neighbors. A node using SLS also periodically receives a location packet from one of its neighbors. The node then updates its location table retaining the most recently received table entries. See [12] for further details on SLS.

There are some similarities of SLS [12] to the Internet standard Routing Information Protocol (RIP) [29]. Specifically, both transmit tables to neighbors on a periodic basis. Differences between these two protocols include the following: SLS sends partial location tables compared to the total routing tables sent by RIP and, unlike RIP, a node using SLS will utilize its current location table in the calculation of its new location table. Lastly, RIP was not developed for an ad hoc network environment. Thus, SLS shares its tables with different neighbors as the network topology changes and, like OSPF, the rate of RIP update packets is only based on time not on distance moved.

3.3 Legend Exchange and Augmentation Protocol (LEAP)

The Legend Exchange and Augmentation Protocol (LEAP) [36] is another proactive location dissemination system. LEAP consists of a legend (i.e., an explanatory list) that migrates (or leaps) from one node to another node in a heterogeneous network. The legend's current state is the locations of nodes that have been previously collected. As the legend traverses the network, it collects the location information of each node, and distributes this information to all other nodes in the network.

In LEAP, each node in the ad hoc network periodically broadcasts a packet to its neighbors to announce its existence [36]. This broadcast packet or "hello" packet includes the current node's location information and the time the packet was sent.

There are two types of location tables in LEAP [36]. A local location table is stored in every node and includes location information on other nodes. The legend is a global location table and includes location information on all nodes as well as information to decide where to send the legend next. Initially, each node in the MANET has an empty local location table that can store *n* entries for the *n* nodes in the network. As mentioned in Section 1.2, each entry in a node's local location table has the following items for each node in the network:

- ID — node ID,
- loc_info — location information for the node, and
- last_update_time — time stamp for the location information.

Each entry in the global location table (or legend) includes the previous three items and the following item:

- v_bit — boolean parameter to show whether the node has been visited by the legend.

We note that while *n* local location tables exist, only one global location table (or legend) exists.

A local location table is updated in two ways [36]. First, every time a node receives a "hello" packet, the node updates corresponding entries for its neighbors in its local location table. Second, when the legend visits a node, both the global location table and the local location table are updated based on the timestamps for the corresponding entries in the two tables. In other words, for each entry, the most recent entry of one table is stored in the other table. After this update procedure finishes, the legend migrates (or leaps) to another node. Thus, at this time, both the most recently visited node and the legend have identical location information for all visited nodes.

In the first implementation of LEAP [36], only one legend in the network exists. However, the authors state that they plan to consider the effect of multiple legends traversing a flat network. They also plan to develop a *hierarchical* legend-based service that provides current information (e.g., location information) on the state of nodes in a hierarchical network.

Initially, one node is selected to begin the single legend propagation [36]. That node sets, for each node in the global location table, the visited bit to false and the current location of the node to undefined as an initialization step. The node then updates its current location and location of all known neighbors in the global location table (with a timestamp), sets its visited bit in the global location table, and then sends the legend to the closest un-visited neighbor. If all neighbors of the current node have been visited, the node sends the legend to the closest un-visited node using the Location-Aided Routing (LAR) protocol [44].

It is possible that the location information of some nodes are unknown by the currently visited nodes due to a partitioned network. Thus, LEAP can not tell which node is the legend's closest un-visited node [36]. In this case, LEAP chooses the lowest ID node as the destination for the legend's next migration. If a partitioned network exists, the destination for the legend may not be reachable. When this problem occurs, the legend waits on the current node for a timeout period. When the timeout expires, the destination may now be reachable due to a change in the topology of the network. If the destination is still unreachable, LEAP repeats this wait procedure a second time. If the node is still unreachable, LEAP chooses a new destination.

After all nodes in the global location table are visited, the legend is paused at the last visited node for a timeout period [36]:

$$\left(\frac{T_{range}}{\alpha}\right) * \left(\frac{1}{\nu}\right) = \frac{T_{range}}{\alpha\nu},$$

where T_{range} is the transmission range of the nodes, ν is the average velocity of the nodes, and α is a scaling factor. That is, the time the legend is paused corresponds to the movement of the nodes. If the nodes are moving quickly, then the legend propagates (almost) continuously; if the nodes are moving slowly, then the legend is often paused. In other words, there is no reason to propagate the legend if the locations of the nodes have barely changed. When the timeout expires, the visited bit of all nodes in the global location table is set to false. A new propagation of the legend then continues in the same manner.

In a real network environment, the legend may get lost during transmission [36]. A node sending the legend sets an acknowledgment timer to help ensure the reliable transmission of the legend. When node A forwards the legend packet to node B, node A sets a timer to receive a response from node B. If node B is a neighbor, the response is over-hearing node B propagate the legend further. If node B is not a neighbor, then node B sends an acknowledgment packet to node A. Node B also sends an acknowledgment packet to node A if the legend is to be paused at node B. If node A receives a response from node B, then node A cancels the timer; otherwise, node A attempts to send the legend to node B a second time. If the timer expires a second time, node A chooses a new destination to send the legend packet.

A legend traversal example is illustrated in Figure 8. (This figure is taken from [36].) We note that this figure is not a snapshot at a certain time. Instead, Figure 8 is a record of the position for each node when it receives the legend. The legend begins its traversal at node 0 and ends its traversal at node 28. The dotted lines illustrate the use of LAR to transmit the legend to a non-neighbor node (e.g., node 3 sending the legend to node 7). As shown, LAR is needed to forward the legend three times in this example. The LAR case for node 4 is different from the other two LAR cases. After the legend visits node 42, which is at position (15,225), it determines node 4 is the next destination. However, it's impossible for node 42 to transfer any packet to node 4 as the network is partitioned. Thus, LEAP suspends the legend propagation for a timeout. When node 4 has moved from location 4 to 4' (in Figure 8), the network is again connected and node 42 can send node 4 the legend.

3.4 Ants

Mobile software agents, modeled on ant behavior, are used to collect and disseminate location information in GPSAL (the GPS/Ant-Like routing algorithm) [14]. An ant is an agent sent by a source node to a destination node in the network. The destination node may be the node farthest from the source, the node with the oldest location information in the source's routing table, or a random node. Similar to LEAP [36], as the ant traverse to its destination, intermediate nodes update their location tables with more recent location information. In addition, if an intermediate node has more recent location information, it modifies the ant packet. Thus, the main difference between LEAP and the location service in GPSAL is the method that is used to create/propagate

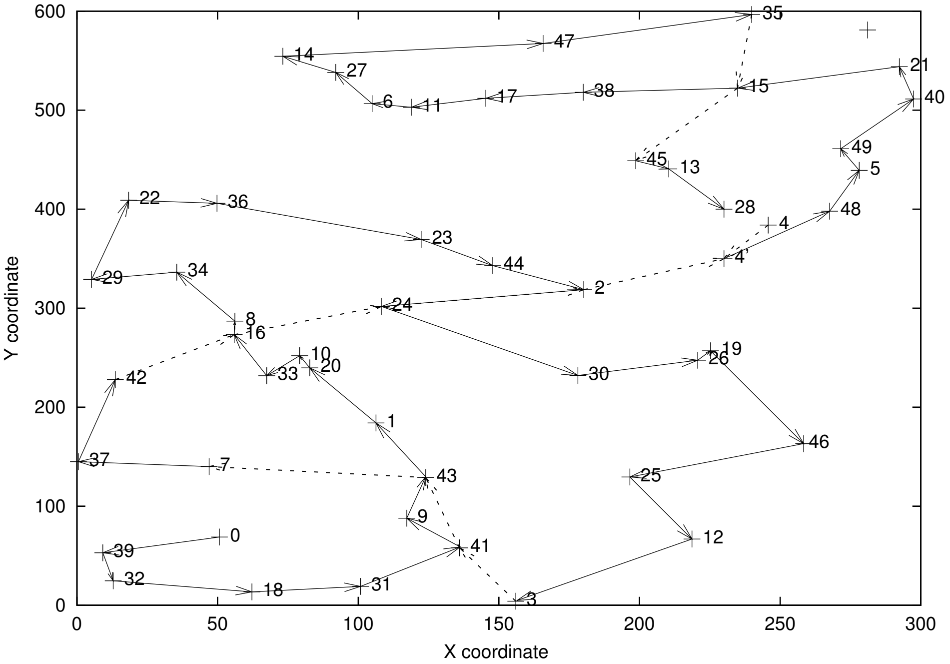

Figure 8: Example of the legend in LEAP migrating.

the legend/ant. Ants in [14] are associated with specific nodes; the legend in LEAP [36] is a global entity.

3.5 Geographic Region Summary Service (GRSS)

The Geographic Region Summary Service (GRSS) is proposed in [32, 33]. Like GLS, the area covered by the ad hoc network is arranged into a hierarchy of grids with squares of increasing size (see Figure 4). The location service is based upon two main functions. First, a node within the same order-0 square[3] transmits its location and neighborhood information to all other nodes in the same order-0 square. This task is accomplished via a link-state routing protocol. Thus, like GLS, full topology information is known within each order-0 square. Second, boundary nodes can transmit location information to nodes in adjacent squares; thus, boundary nodes in an order-i square share location information for all nodes within the order-i square to sibling order-i squares (four siblings of order-i make a square of order-$i+1$) via "summary" location packets.

A summary packet is an aggregated list of location information [32, 33]. Specifically, a summary packet contains a list of all nodes within the order-i square and the location of the order-i square's center. In other words, the hierarchical infrastructure created is used to generate and disseminate summaries that contain location information. The union of all received summary packets and the location information on a given node's order-0 square contains location information

---

[3] We note that GLS begins counting at one instead of zero.

for all nodes in the network.

One advantage of GRSS is that a given node does not flood its location information in the entire network [32, 33]. One disadvantage of GRSS is that, due to summary packets, location information may only be the center of the region a node resides in instead of the exact location of the node. However, since the size of each region is approximately the radio range, all nodes in an order-0 square can be reached within two hops [32].

The authors of [32] propose two types of summary packets: exact summary and imprecise summary. Exact summaries require more overhead than imprecise summaries, but then lead to fewer false positives. A false positive occurs when a node receives conflicting location information on another node, which causes a packet to be dropped. See [32] for more details on GRSS.

3.6 Dead Reckoning: A Location Prediction Technique

A recently proposed location dissemination location system is the the Dead Reckoning Method (DRM) [1, 45]. In DRM, each node constructs a model of its movement which is then disseminated with the node's current location. Other nodes are then able to predict the movement of every other node in the network. The basis for this technique is the prediction method discussed in [47].

The authors of DRM mention that the movement model constructed by the nodes could be a deterministic or probabilistic first order or high order model [1]. In [1, 45], the following movement model is used. A node computes its velocity values (v_x and v_y) from two successive location samples, (x_1, y_1) and (x_2, y_2), taken at times t_1 and t_2:

$$v_x = \frac{x_2 - x_1}{t_2 - t_1} \qquad \text{and} \qquad v_y = \frac{y_2 - y_1}{t_2 - t_1}.$$

Once the velocity values are computed, the node floods the values, the node's current location, and the time the velocity values were computed. The packet transmitted is called a DRM update packet.

A node predicts another node's location via the following two equations [1, 45]:

$$x_{predict} = x_{location} + (v_{x_{model}} * (t_{current} - t_{model}))$$
$$\text{and}$$
$$y_{predict} = y_{location} + (v_{y_{model}} * (t_{current} - t_{model})),$$

where $(x_{location}, y_{location})$, $(v_{x_{model}}, v_{y_{model}})$, and t_{model} are the values received in the received DRM update packet and $t_{current}$ is the current time.

Each node periodically predicts its own location as well, which is compared against the node's actual value. That is, a deviation between a node's actual location and predicated location is calculated by:

$$d = \sqrt{(x_{current} - x_{predicted})^2 + (y_{current} - y_{predicted})^2}.$$

If its own predicted location deviates from its actual location by a given threshold, then the node transmits a new DRM update packet. Thus, if a node transmits an accurate movement model, very little location update cost exists in DRM [1, 45].

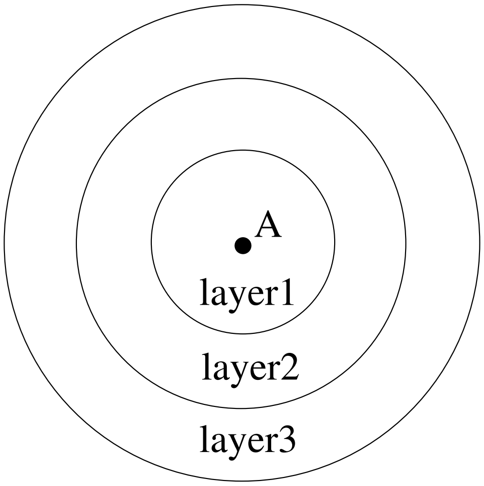

Figure 9: The layering method in DRM.

In [1], the authors mention an optimization to their dead reckoning technique that is based on the "distance effect" discussed in Section 3.1. Specifically, a node floods a DRM update packet if the calculated deviation is large; if the calculated distance is small, a node only transmits the DRM update packet to other nodes in close proximity of the node. In [45], the authors detail a layering method, based on a given node's location (see Figure 9), to implement this optimization.

Once a node can predict all other node locations, [1] proposes that each node computes the topology of the network. Dijkstra's shortest path algorithm can then be used to route packets from a source to a destination. In [45], the authors use geographic forwarding to route packets.

4 Reactive Location Systems

The last type of location systems surveyed are reactive in nature. As mentioned in Section 1.2, reactive location services query location information on an as needed basis. In this section, we survey three reactive location systems proposed in the literature. According to the *Mauve Classification* [51], reactive location systems would be classified as an *all-for-some* approach. That is, every node in the network maintains location information on some other nodes in the network whose locations are needed.

4.1 Reactive Location Service (RLS)

In the Reactive Location Service (RLS) [12], when a mobile node requires a location for another node and the location information is either unknown or expired, the requesting node will first ask its neighbors for the requested location information (i.e., the time-to-live, or TTL, of the packet is set to zero). If the node's neighbors do not respond to the requested location information within a timeout period, then the node will flood a location request packet in the entire network.

When a node receives a location request packet and does not know the requested location information, the node propagates the flooded location request. If, however, a node receives a location request packet and the node's location table contains the requested location information, the node returns a location reply packet via the reverse source route obtained in the location request packet. In other words, each location request packet carries the full route (a sequenced list of nodes) that a location reply packet should be able to traverse in its header. Since IEEE 802.11 requires bi-directional links in the delivery of all non-broadcast packets, RLS assumes bi-directional links. If bi-directional links are not available, this requirement can be removed via the manner proposed in the Dynamic Source Routing (DSR) protocol [10].

If feasible, each node using RLS should update its location table when a new location packet is overheard/received. In other words, RLS suggests promiscuous mode operation is used by all nodes. (As noted in [37], promiscuous mode operation is power consuming.) Lastly, entries in the location table are aged as the node associated with the entry moves; that is, an entry associated with a node that is moving quickly will age more quickly.

There are some similarities of RLS to LAR [44] and DSR [10], two unicast routing protocols developed for a mobile ad hoc network. Specifically, all three protocols are reactive protocols that try to discover the required information on demand. In addition, all three protocols use the reverse source route to respond to a request for information. One significant difference between these three protocols is the following: RLS attempts to determine location information, while LAR and DSR attempt to determine full routes. Lastly, requesting desired information from neighbors first and allowing intermediate nodes to reply to a request are two features that both RLS and DSR have. Although not mentioned in LAR [44], the usefulness of these features for LAR are evaluated in [13].

4.2 Reactive Location Service (RLS$'$)

Another RLS protocol, which was developed concurrently, is proposed in [42] and then used for routing in a city environment in [50]. To avoid confusion, we refer to this second RLS protocol as RLS$'$. The differences between RLS and RLS$'$ are:

- While intermediate nodes are allowed to reply to location requests in RLS [12], a location request is forwarded until it reaches the node whose location is desired in RLS$'$ [42, 50]. While cached replies are discussed in [42], the authors chose not to implement them; the authors felt the disadvantages (e.g., several location replies may be returned for one location request) outweighed the advantages.

- A location reply packet is returned via the reverse source route in RLS [12]; a location replay packet is returned via "the underlying routing protocol (e.g. greedy unicast routing, flooding, etc.)" in RLS$'$ [42].

- While several options for flooding a location request packet are discussed in RLS$'$ [42] (e.g., linear flooding and exponential flooding), the main flooding scheme evaluated is the same as the scheme used in RLS [12]. This scheme is called binary flooding in [42].

- RLS$'$ includes radial flooding, which allows nodes farther away from a source to rebroadcast a location request packet before nodes closer to the source [42]. RLS does not include this feature.

4.3 LOcation Trace Aided Routing (LOTAR)

LOTAR (LOcation Trace Aided Routing) [69], which is a routing protocol similar to LAR, has neighbors periodically exchange location tables. With more current location information, the request and expected zones of LAR are more accurately defined.

5 Conclusions and Future Research

We survey 17 different location information services that exist in the literature to date in this chapter. An effective location service can be used to improve the performance and scalability of a routing protocol that requires location information (e.g., GPSR [40]). We classify the location information services into three categories: proactive location database systems, proactive location dissemination systems, and reactive location systems. Proactive location services are those protocols that have nodes exchange location information periodically; the information is either exchanged with a few select nodes or with all nodes. Reactive location services query location information on an as needed basis.

There are several areas of future investigation on location services that are needed. First, as mentioned in [51], when a node is associated with a position, location privacy is difficult to achieve. Since none of the surveyed location services consider anonymity, future research in this area is essential.

In addition, more quantitative comparisons to discover the strengths and weaknesses of the various approaches are needed. Typically a performance evaluation is given in a paper that proposes a location service. Seldom, however, are multiple location services quantitatively compared. The exceptions are:

- In [51], the authors state the complexity and robustness of four location services (i.e., DLS, home regions, UQS, and GLS).
- A detailed comparison of DLS, SLS, and RLS exists in [12].
- RLS$'$ with greedy location-based forwarding is compared to DSR and GLS with greedy location-based forwarding in [42].
- A comparison of LEAP, GLS, RLS, and SLS is given in [36],
- In [45], a performance comparison exists for a geographical forwarding routing protocol that uses four different location services: DRM, GRSS, DLS, and SLS.

According to [43], GLS is a promising distributed location service. However, due to its complexity, GLS requires a "deeper analysis and evaluation" [43]. See [25], [36], and [43] for further performance results on GLS.

6 Acknowledgements

We thank several members (past and present) of Toilers (`http://toilers.mines.edu`) for the information presented herein, especially Jeff Boleng, Lucas Wilcox, Nalinrat Guba, and Xia Jiang.

References

[1] A. Agarwal and S.R. Das. Dead reckoning in mobile ad hoc networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2003)*, pages 1838–1843, 2003.

[2] K.N. Amouris, S. Papavassiliou, and M. Li. A position-based multi-zone routing protocol for wide area mobile ad-hoc networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, pages 1365–1369, 1999.

[3] S. Basagni, I. Chlamtac, and V.R. Syrotiuk. Geographic messaging in wireless ad hoc networks. In *IEEE Vehicular Technology Conference*, pages 1957–1961, 1999.

[4] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 76–84, 1998.

[5] L. Blăzević, L. Buttyán, S. Čapkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec. Self organization in mobile ad hoc networks: the approach of Terminodes. *IEEE Communications Magazine*, 39(6):166–174, June 2001.

[6] L. Blăzević, S. Giordano, and J.-Y. Le Boudec. Self organized terminode routing. *Cluster Computing Journal*, 5(3), April 2002.

[7] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, pages 48–55, 1999.

[8] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *ACM Wireless Networks*, 7(6):609–616, November 2001.

[9] A. Boukerche and S. Rogers. GPS query optimization in mobile and wireless ad hoc networking. In *Proceedings of the 6th IEEE Symposium on Computers and Communications*, pages 198–203, 2001.

[10] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. Multi-hop wireless ad hoc network routing protocols. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 85–97, 1998.

[11] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, October 2000.

[12] T. Camp, J. Boleng, and L. Wilcox. Location information services in mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC))*, pages 3318–3324, 2001.

[13] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi. Performance evaluation of two location based routing protocols. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1678–1687, 2002.

[14] D. Câmara and A. Loureiro. A novel routing algorithm for ad hoc networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, volume 8, pages 1–8, 2000.

[15] S. Čapkun, M. Hamdi, and J. P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, volume 9, pages 1–10, 2001.

[16] S. Čapkun, M. Hamdi, and J. P. Hubaux. GPS-free positioning in mobile ad-hoc networks. *Cluster Computing Journal*, 5(2), April 2002.

[17] S. Das, R. Castañeda, and J. Yan. Simulation-based performance evaluation of routing protocols for mobile ad hoc networks. *ACM Mobile Networks and Applications*, 5(3):179–189, 2000.

[18] S. Das, C. Perkins, and E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 3–12, 2000.

[19] S. Datta, I. Stojmenovic, and J. Wu. Internal node and shortcut based routing with guaranteed delivery in wireless networks. In *Proceedings of the IEEE International Conference on Distributed Computing Systems Workshop*, pages 461–466, 2001.

[20] D. De Couto and R. Morris. Location proxies and intermediate node forwarding for practical geographic forwarding. Technical Report MIT-LCS-TR-824, MIT Laboratory for Computer Science, June 2001.

[21] G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical report, USC/Information Sciences Institute, ISI/RR-87-180, March 1987.

[22] S. Giordano and M. Hamdi. Mobility management: The virtual home region. Technical Report SSC/1999/037, Ecole Polytechnique Federale de Lausanne, October 1999.

[23] S. Giordano and I. Stojmenovic. Position based ad hoc routes in ad hoc networks. In M. Ilyas, editor, *Handbook of Ad Hoc Wireless Networks*. CRC Press, 2003.

[24] S. Giordano, I. Stojmenovic, and L. Blăzević. Position based routing algorithms for ad hoc networks: A taxonomy. In X. Cheng, X. Huang, and D.Z. Du, editors, *Ad Hoc Wireless Networking*. Kluwer Academic Publishers, 2003.

[25] N. Guba and T. Camp. Recent work on GLS: a location service for an ad hoc network. In *Proceedings of the Grace Hopper Celebration (GHC 2002)*, 2002.

[26] Z. Haas. A new routing protocol for reconfigurable wireless networks. In *Proceedings of the IEEE International Conference on Universal Personal Communications*, Oct. 1997.

[27] Z.J. Haas and B. Liang. Ad hoc mobility management with randomized database groups. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 1756–1762, 1999.

[28] Z.J. Haas and B. Liang. Ad hoc mobility management with uniform quorum systems. *IEEE/ACM Transactions on Networking*, 7(2):228–240, April 1999.

[29] C. Hedrick. Routing Information Protocol. Request for Comments 1058, June 1988.

[30] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[31] T.-C. Hou and V. Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications*, 34(1):38–44, January 1986.

[32] P.-H. Hsiao. Geographical region summary service for geographical routing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):25–39, October 2001.

[33] P.-H. Hsiao. Geographical region summary service for geographical routing. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 263–266, 2001.

[34] J.-P. Hubaux, J.-Y. Le Boudec, S. Giordano, M. Hamdi, L. Blăzević, L. Buttyán, and M. Vojnović. Towards mobile ad-hoc wans: terminodes. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1052–1059, 2000.

[35] R. Jain, A. Puri, and R. Sengupta. Geographical routing using partial information for wireless ad hoc networks. *IEEE Personal Communications*, pages 48–57, February 2001.

[36] X. Jiang and T. Camp. An efficient location server for an ad hoc network. Technical report, Department of Math. and Computer Sciences, Colorado School of Mines, MCS-03-06, May 2003.

[37] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Routing protocols for mobile ad-hoc networks - a comparative performance analysis. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 195–206, 1999.

[38] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imelinsky and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[39] D. Johnson, D. Maltz, Y. Hu, and J. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft: draft-ietf-manet-dsr-04.txt, November 2000.

[40] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 243–254, 2000.

[41] G. Karumanchi, S. Muralidharan, and R. Prakash. Information dissemination in partitionable mobile ad hoc networks. In *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, pages 4–13, 1999.

[42] M. K*ä*semann, H. F*ü*ßler, H. Hartenstein, and M. Mauve. A reactive location service for mobile ad hoc networks. Technical report, Department of Science, University of Mannheim, TR-02-014, November 2002.

[43] M. K*ä*semann, H. Hartenstein, H. F*ü*ßler, and M. Mauve. Analysis of a location service for position-based routing in mobile ad hoc networks. In *Proceedings of the 1st German Workshop on Mobile Ad-Hoc Networks (WMAN)*, pages 121–133, 2002.

[44] Y. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 66–75, 1998.

[45] V. Kumar and S.R. Das. Performance of dead reckoning-based location service for mobile ad hoc networks. *Wireless Communications and Mobile Computing Journal*, 2003.

[46] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 120–130, 2000.

[47] B. Liang and Z.J. Haas. Predictive distance-based mobility management for pcs networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM)*, pages 1377–1384, 1999.

[48] B. Liang and Z.J. Haas. Virtual backbone generation and maintenance in ad hoc network mobility management. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM)*, pages 1293–1302, 2000.

[49] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000.

[50] C. Lochert, H. Hartenstein, J. Tian, H. F*üß*ler, D. Hermann, and M. Mauve. A routing strategy for vehicular ad hoc networks in city environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 156–161, 2003.

[51] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network*, 15(6):30–39, November/December 2001.

[52] J. Moy. OSPF version 2. Request for Comments 2178, July 1997.

[53] G. Pei and M. Gerla. Mobility management for hierarchical wireless networks. *Mobile Networks and Applications*, 6(4):331–337, August 2001.

[54] G. Pei, M. Gerla, X. Hong, and C. Chiang. A wireless hierarchical routing protocol with group mobility. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1538–1542, September 1999.

[55] C. Perkins. IP mobility support. Request for Comments 2002, October 1996.

[56] C. Perkins, E. Royer, and S. Das. Ad hoc on demand distance vector (AODV) routing. Internet Draft: draft-ietf-manet-aodv-07.txt, November 2000.

[57] C. Perkins, E. Royer, S. Das, and M.K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications*, 8(1):16–28, 2001.

[58] C. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[59] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report 3898, INRIA - Rapport de recherche, 2000.

[60] E. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine*, pages 46–55, April 1999.

[61] I. Stojmenovic. A scalable quorum based location update scheme for routing in ad hoc wireless networks. Technical Report TR-99-09, University of Ottawa, September 1999.

[62] I. Stojmenovic. Home agent based location update and destination search schemes in ad hoc wireless networks. In A. Zemliak and N.E. Mastorakis, editors, *Advances in Information Science and Soft Computing*, pages 6–11. WSEAS Press, 2002. Also Technical Report, University of Ottawa, TR-99-10, September 1999.

[63] I. Stojmenovic. Position based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, July 2002.

[64] I. Stojmenovic, M. Russell, and B. Vukojevic. Depth first search and location based localized routing and qos routing in wireless networks. In *Proceedings of the IEEE International Conference on Parallel Processing*, pages 173–180, 2000.

[65] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, March 1984.

[66] Y.C. Tseng, W.H. Liao, and S.L. Wu. Mobile ad hoc networks and routing protocols. In I. Stojmenovic, editor, *Handbook of Wireless Networks and Mobile Computing*, pages 371–392. John Wiley & Sons, 2002.

[67] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 194–205, 2002.

[68] S.-C. Woo and S. Singh. Scalable routing protocol for ad hoc networks. *ACM Wireless Networks*, 7(5):513–529, September 2001.

[69] K. Wu and J. Harms. Location trace aided routing in mobile ad hoc networks. In *Proceedings of the International Conference on Computer Communications and Networks (ICCCN)*, pages 354–359, 2000.