A Survey on Position-Based Routing in Mobile Ad Hoc Networks

Martin Mauve and Jörg Widmer, University of Mannheim
Hannes Hartenstein, NEC Europe, Heidelberg

Abstract

We present an overview of ad hoc routing protocols that make forwarding decisions based on the geographical position of a packet's destination. Other than the destination's position, each node need know only its own position and the position of its one-hop neighbors in order to forward packets. Since it is not necessary to maintain explicit routes, position-based routing does scale well even if the network is highly dynamic. This is a major advantage in a mobile ad hoc network where the topology may change frequently. The main prerequisite for position-based routing is that a sender can obtain the current position of the destination. Therefore, recently proposed location services are discussed in addition to position-based packet forwarding strategies. We provide a qualitative comparison of the approaches in both areas and investigate opportunities for future research.

In recent years the widespread availability of wireless communication and handheld devices has stimulated research on self-organizing networks that do not require a preestablished infrastructure. These *ad hoc networks*, as they are commonly called, consist of autonomous nodes that collaborate in order to transport information. Usually, these nodes act as end systems and routers at the same time.

Ad hoc networks can be subdivided into two classes: *static* and *mobile*. In static ad hoc networks the position of a node may not change once it has become part of the network. Typical examples are rooftop networks [1]. For the remainder of this work we will solely focus on mobile ad hoc networks.

In mobile ad hoc networks, systems may move arbitrarily. Examples where mobile ad hoc networks may be employed are the establishment of connectivity among handheld devices or between vehicles. Since mobile ad hoc networks change their topology frequently and without prior notice, routing in such networks is a challenging task. We distinguish two different approaches: *topology-based* and *position-based* routing.

Topology-based routing protocols use the information about the links that exist in the network to perform packet forwarding. They can be further divided into *proactive*, *reactive*, and *hybrid* approaches.

Proactive algorithms employ classical routing strategies such as distance-vector routing (e.g., DSDV [2]) or link-state routing (e.g., OLSR [3] and TBRPF [4]). They maintain routing information about the available paths in the network even if these paths are not currently used. The main drawback of these approaches is that the maintenance of unused paths may occupy a significant part of the available bandwidth if the topology of the network changes frequently [5].

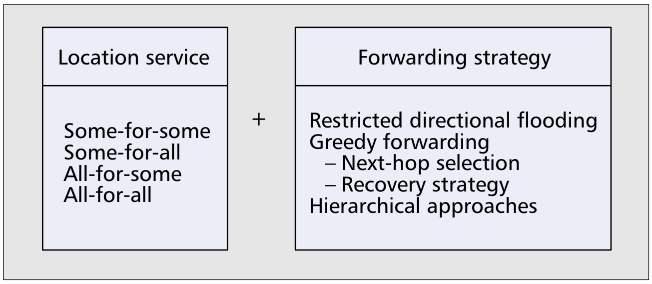
In response to this observation, reactive routing protocols were developed (e.g., DSR [6], TORA [7], and AODV [8]). Reactive routing protocols maintain only the routes that are currently in use, thereby reducing the burden on the network when only a small subset of all available routes is in use at any time. However, they still have some inherent limitations. First, since routes are only maintained while in use, it is typically required to perform a route discovery before packets can be exchanged between communication peers. This leads to a delay for the first packet to be transmitted. Second, even though route maintenance for reactive algorithms is restricted to the routes currently in use, it may still generate a significant amount of network traffic when the topology of the network changes frequently. Finally, packets en route to the destination are likely to be lost if the route to the destination changes.

Hybrid ad hoc routing protocols such as ZRP [9] combine local proactive routing and global reactive routing in order to achieve a higher level of efficiency and scalability. However, even a combination of both strategies still needs to maintain at least those network paths that are currently in use, limiting the amount of topological changes that can be tolerated within a given amount of time. A survey and comparison of topology-based approaches can be found in [10, 11]. In the following we will focus exclusively on position-based routing.

Position-based routing algorithms eliminate some of the limitations of topology-based routing by using additional information. They require that information about the physical position of the participating nodes be available. Commonly, each node determines its own position through the use of GPS or some other type of positioning service [12, 13], a survey of these methods can be found in [14]. A *location service* is used by the sender of a packet to determine the position of the destination and to include it in the packet's destination address.

The routing decision at each node is then based on the destination's position contained in the packet and the position of the forwarding node's neighbors. Position-based routing thus does not require the establishment or maintenance of routes. The nodes have neither to store routing tables nor to transmit messages to keep routing tables up to date. As a further advantage, position-based routing supports the delivery of packets to all nodes in a given geographic region in a natural way. This type of service is called *geocasting* [15].

 0890-8044/01/$10.00 © 2001 IEEE



■ Figure 1. *Building blocks and criteria for classification.*

In this article we present a survey of position-based routing for mobile ad hoc networks. We outline the main problems that have to be solved for this class of routing protocols and present the solutions currently available.

The remainder of this article is structured as follows: We present the basic idea of position-based addressing and routing, and give criteria for a taxonomy of the various proposals. We cover techniques for location services and outline position-based forwarding strategies. A later section contains a qualitative comparison of the location services, and forwarding strategies. We point out open issues and possible directions of future research, and then conclude the article.

Basic Principles and Problems

Before a packet can be sent, it is necessary to determine the position of its destination. Typically, a location service is responsible for this task. Existing location services can be classified according to how many nodes host the service. This can be either *some* specific nodes or *all* nodes of the network. Furthermore, each location server may maintain the position of *some* specific or *all* nodes in the network. We abbreviate the four possible combinations as some-for-some, some-for-all, all-for-some, and all-for-all in the discussion of location services.

In position-based routing, the forwarding decision by a node is primarily based on the position of a packet's destination and the position of the node's immediate one-hop neighbors. The position of the destination is contained in the header of the packet. If a node happens to know a more accurate position of the destination, it may choose to update the position in the packet before forwarding it. The position of the neighbors is typically learned through one-hop broadcasts. These beacons are sent periodically by all nodes and contain the position of the sending node.

We can distinguish three main packet forwarding strategies for position-based routing: greedy forwarding, restricted directional flooding, and hierarchical approaches.

For the first two, a node forwards a given packet to one (greedy forwarding) or more (restricted directional flooding) one-hop neighbors that are located closer to the destination than the forwarding node itself. The selection of the neighbor in the greedy case depends on the optimization criteria of the algorithm. We will present the diverse strategies that existing algorithms use to make this selection.

It is fairly obvious that both forwarding strategies may fail if there is no one-hop neighbor that is closer to the destination than the forwarding node itself. *Recovery strategies* that cope with this kind of failure are also discussed in a later section.

The third forwarding strategy is to form a hierarchy in order to scale to a large number of mobile nodes. In this article we investigate two representatives of hierarchical routing that use greedy forwarding for wide area routing and non-position-based approaches for local area routing. Figure 1 illustrates the two building blocks — location service and forwarding strategy — required for position-based routing, together with classification criteria for the various existing approaches.

Location Services

In order to learn the current position of a specific node, the help of a *location service* is needed. Mobile nodes register their current position with the service. When a node does not know the position of a desired communication partner, it contacts the location service and requests that information. In classic cellular networks, there are dedicated position servers (with well-known addresses) that maintain position information about the nodes in the network. With respect to the classification, this is a some-for-all approach since the servers are *some* specific nodes, each maintaining position information about *all* mobile nodes.

In mobile ad hoc networks, such a centralized approach is viable only as an external service that can be reached via non-ad-hoc means. There are two main reasons for this. First, it would be difficult to obtain the position of a position server if the server were part of the ad hoc network itself. This would represent a chicken-and-egg problem: without a position server, it is not possible to get position information, but without the position information the server cannot be reached. Second, since an ad hoc network is dynamic, it might be difficult to guarantee that at least one position server will be present in a given ad hoc network.

In the following we concentrate on decentralized location services that are part of the ad hoc network.

Distance Routing Effect Algorithm for Mobility

Within the Distance Routing Effect Algorithm for Mobility (DREAM) framework [16], each node maintains a position database that stores position information about each other node that is part of the network. It can therefore be classified as an all-for-all approach. An entry in the position database includes a node identifier, the direction of and distance to the node, as well as a time value that indicates when this information was generated. Of course, the accuracy of such an entry depends on its age.

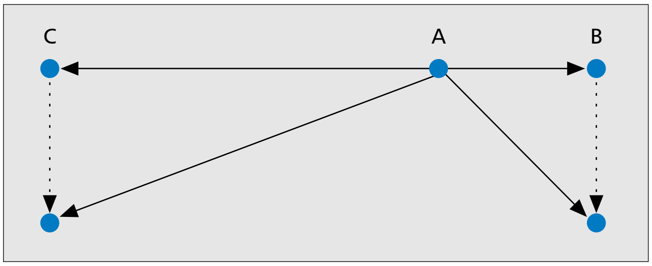
Each node regularly floods packets to update the position information maintained by the other nodes. A node can control the accuracy of its position information available to other nodes by:

- The frequency at which it sends position updates (*temporal resolution*)
- Indicating how far a position update may travel before it is discarded (*spatial resolution*)

The temporal resolution of sending updates is coupled with the mobility rate of a node (i.e., the higher the speed, the more frequent the updates). The spatial resolution is used to provide accurate position information in the direct neighborhood of a node and less accurate information at nodes farther away. The costs associated with accurate position information at very remote nodes can be reduced since, as the authors argue, "the greater the distance separating two nodes, the slower they appear to be moving with respect to each other" (termed the *distance effect* [17]). An example of this "distance effect" is given in Fig. 2. Assume that in this example node A is not moving, while nodes B and C are moving in the same direction at the same speed. From node A's perspective, the change in direction will be greater for node B than for node C. The distance effect allows low spatial resolution in areas far away from the target node, provided that intermediate hops are able to update the position information contained in the packet.

Quorum-Based Location Service

The concept of *quorum systems* is well known from information replication in databases and distributed systems. Information updates (write operations) are sent to a subset (quorum)



■ Figure 2. *The distance effect.*

of available nodes, and information requests (read operations) are referred to a potentially different subset. When these subsets are designed such that their intersection is nonempty, it is ensured that an up-to-date version of the sought-after information can always be found.

In [18], this scheme is used to develop a location service for ad hoc networks. We will discuss it by means of the simple sample network shown in Fig. 3. A subset of all mobile nodes is chosen to host position databases; in the example, these are nodes 1–6. A virtual backbone is constructed between the nodes of the subset, using a non-position-based ad hoc routing mechanism.

A mobile node sends position update messages to the nearest backbone node, which then chooses a quorum of backbone nodes to host the position information. Thus, node D sends its updates to node 6, which might then select quorum A with the nodes 1, 2, and 6 to host the information. When a node S wants to obtain the position information, it sends a query to the nearest backbone node, which in turn contacts the nodes of a (usually different) quorum.[1] Node 4 might, for example, choose quorum B, consisting of nodes 4, 5, and 6, for the query. Since by definition the intersection of two quorums is non-empty, the querying node is guaranteed to obtain at least one response with the desired position information.

It is important to timestamp position updates, since some nodes in the queried quorum might have been in the quorum of previous updates and would then report outdated position information. If several responses are received, the one representing the most current position update is chosen.

An important aspect of quorum-based position services is the following trade-off: the larger the quorum sets, the higher the cost for position updates and queries, but also the larger the number of nodes in the intersection of two quorums, which improves resilience against unreachable backbone nodes. In [18] several methods on how to generate quorum systems with the desired properties are discussed. In the article, the authors also show that the size of the quorum can be kept independent of the number of nodes by dividing the nodes into sub-sets of a constant size. An individual virtual backbone is constructed for each of these subsets.

The quorum-based position service can be configured to operate as all-for-all, all-for-some, or some-for-some approach, depending on how the size of the backbone and the quorum is chosen. However, it will typically work as a some-for-some scheme with the backbone being a small subset of all available nodes and a quorum being a small subset of the backbone nodes.

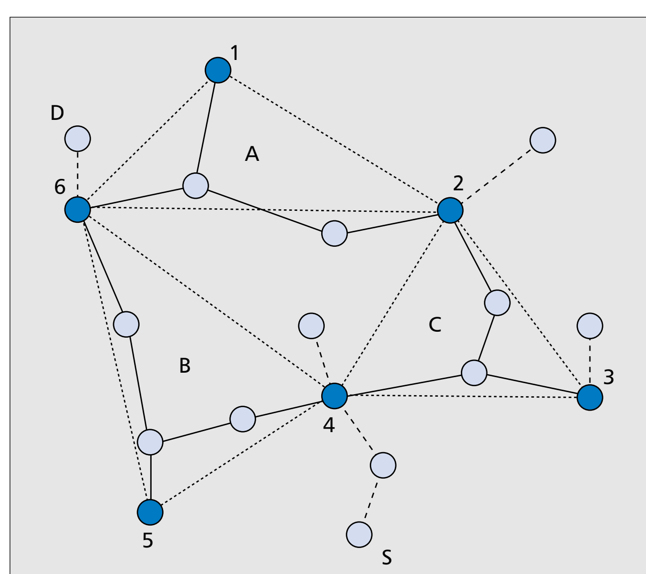
Other work based on quorums is presented in [19]. Here, position information for the nodes is propagated in a north-south direction. Whenever a node whose position is unknown has to be contacted, position information is searched in east-west direction until the information is found. While the algorithm described is still at an early stage, it is an interesting idea worth further study.

Grid Location Service

The Grid Location Service (GLS) [20, 21] is part of the Grid project [22]. It divides the area that contains the ad hoc network into a hierarchy of squares. In this hierarchy, n-order squares contain exactly four (n – 1)-order squares, forming a so called *quadtree*. Each node maintains a table of all other nodes within the local first-order square. The table is constructed with the help of periodic position broadcasts scoped to the area of the first-order square.

Again, we demonstrate the mechanism by means of a simple example (Fig. 4). To determine where to store position information, GLS establishes a notion of *near* node IDs, defined as the least ID greater than a node's own ID.[2] When node 10 in the figure wants to distribute its position information, it sends position updates to the respective node with the nearest ID in each of the three surrounding first-order squares. Thus, the position information is available at the nodes 15, 18, 73 and at all nodes that are in the same first-order square as 10 itself. In the surrounding three second-order squares, again the nodes with the nearest ID are chosen to host the node's position; in the example these are nodes 14, 25, and 29. This process is repeated until the area of the ad-hoc network has been covered. The density of position information for a given node thus decreases logarithmically with the distance from that node.

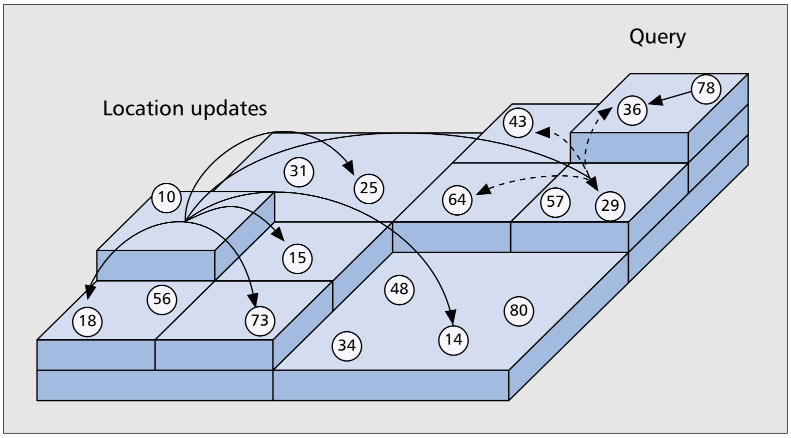
Assume now that node 78 wants to obtain the position of node 10. It should therefore locate a nearby node that knows about the position of node 10. In the example this is node 29. While node 78 does not know that node 29 holds the required position, it is able to discover this information. To see how this process works, it is useful to take a look at the position servers for node 29. Its position is stored in the three surrounding first-order squares at nodes 36, 43, and 64. Note that each of these nodes, as well as node 29, are automatically also the ones in their respective first-order square with the ID nearest to 10. Thus, there exists a "trail" of descending node IDs from each of the squares of *all* orders to the correct position server. Position queries for a node can now be directed to the node with



■ Figure 3. *A quorum.*

[1] *The updates and responses can be multicast to the corresponding servers in case multicast is supported by the backbone's ad hoc routing protocol.*

[2] *ID numbers wrap around after the highest possible ID.*



■ Figure 4. *GLS.*

the nearest ID of which the querying node knows. In our example this would be node 36. The node with the nearest ID does not necessarily know the node sought, but will know a node with a nearer node ID (node 29, which is already the sought-after position server). The process continues until a node that has the position information available is found.

Note that a node need not know the IDs of its position servers, which makes a bootstrapping mechanism to discover a node's position servers unnecessary. Position information is forwarded to a certain position (e.g., the lower left corner) of each element in the quad tree. After reaching a node close to this position, the position information is forwarded progressively to nodes with closer IDs in a process resembling position queries. This ensures that the position information reaches the correct node, where it is then stored.

Since GLS requires that all nodes store the information on some other nodes, it can be classified as an all-for-some approach.

Homezone

Two almost identical location services have been proposed independently [23, 24]. Both use the concept of a virtual *Homezone* where position information for a node is stored. The position *C* of the Homezone for a node can be derived by applying a well-known hash function to the node identifier. All nodes within a disk with radius *R* centered at *C* have to maintain position information for the node. Thus, as in the case of the Grid Position Service, a position database can be found by means of a hash function on which sender and receiver agree without having to exchange information. The Homezone approaches are therefore also all-for-some approaches. If the Homezone is sparsely populated, *R* may have to be increased, resulting in several tries with increasing *R* for updates as well as queries.

Forwarding Strategies

Greedy Packet Forwarding

Using greedy packet forwarding, the sender of a packet includes the approximate position of the recipient in the packet. This information is gathered by an appropriate location service (e.g., one of those described above). When an intermediate node receives a packet, it forwards the packet to a neighbor lying in the general direction of the recipient. Ideally, this process can be repeated until the recipient has been reached.

Generally, there are different strategies a node can use to decide to which neighbor a given packet should be forwarded. These are illustrated in Fig. 5, where S and D denote the source and destination nodes of a packet, respectively. The circle with radius *r* indicates the maximum transmission range of S. One intuitive strategy is to forward the packet to the node that makes the most progress towards (is closest to) D. In the example this would be node C. This strategy is known as *most forward within r* (MFR) [25]; it tries to minimize the number of hops a packet has to traverse in order to reach *D*.

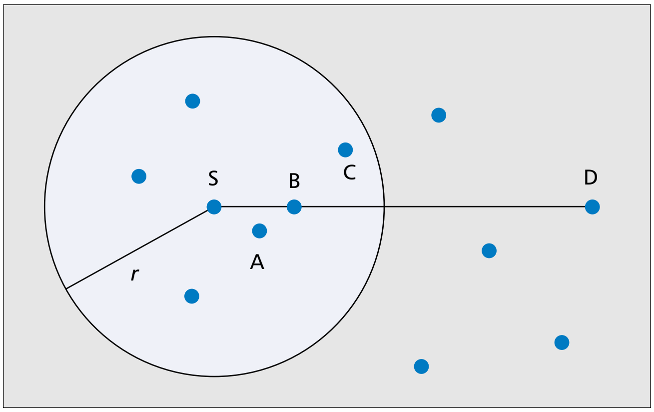
MFR is a good strategy in scenarios where the sender of a packet cannot adapt the signal strength of the transmission to the distance between sender and receiver. However, in [26] it is shown that a different strategy performs better than MFR in situations where the sender can adapt its signal strength. In *nearest with forward progress* (NFP), the packet is transmitted to the nearest neighbor of the sender which is closer to the destination. In Fig. 5 this would be node A. If all nodes employ NFP, the probability of packet collisions is reduced significantly. Therefore, the average progress of the packet, calculated as $p \cdot f(a, b)$ where p is the likelihood of a successful transmission without a collision and $f(a, b)$ is the progress of the packet when successfully forwarded from a to b, is higher for NFP than for MFR.

Another strategy for forwarding packets is compass routing, which selects the neighbor closest to the straight line between sender and destination [27]. In the example this would be node B. Compass routing tries to minimize the spatial distance a packet travels.

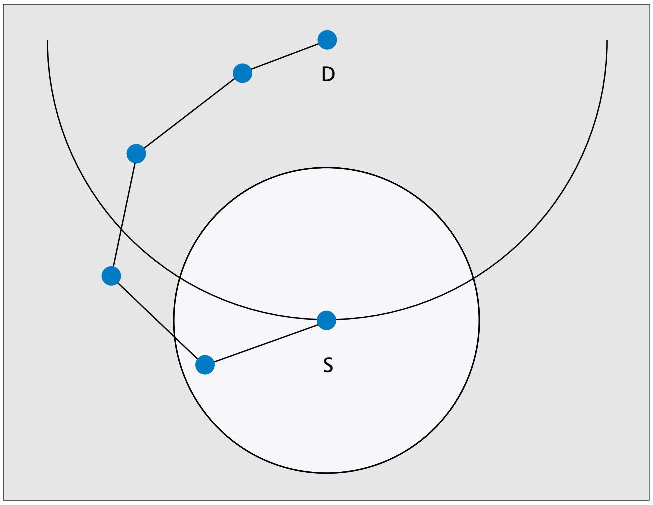
Finally, it is possible to let the sender randomly choose one of the nodes closer to the destination than itself and forward the packet to that node [28]. This strategy minimizes the accuracy of information needed about the position of the neighbors and reduces the number of operations required to forward a packet.

Unfortunately, greedy routing may fail to find a path between sender and destination, even though one does exist. An example of this problem is depicted in Fig. 6. In this figure the half-circle around D has the radius of the distance between S and D, and the circle around S shows the transmission range of S. Note that there exists a valid path from S to D. The problem here is that S is closer to the destination D than any of the nodes in its transmission range. Greedy routing therefore has reached a local maximum from which it cannot recover.

To counter this problem it has been suggested that the packet should be forwarded to the node with the least backward (negative) progress [25] if no nodes can be found in the forward direction. However, this raises the problem of looping packets, which cannot occur when packets are forwarded only



■ Figure 5. *Greedy routing strategies.*



■ Figure 6. *Greedy routing failure.*

toward the destination with positive progress. Other researchers proposed not to forward packets that have reached a local maximum at all [26].

The face-2 algorithm [29] and the perimeter routing strategy of the Greedy Perimeter Stateless Routing Protocol (GPSR) [30, 31] are two very similar recovery approaches based on planar graph traversal. Both are performed on a per-packet basis and do not require nodes to store any additional information. A packet enters the recovery mode when it arrives at a local maximum. It returns to greedy mode when it reaches a node closer to the destination than the node where the packet entered the recovery mode.

Planar graphs are graphs with no intersecting edges. A set of nodes in an ad hoc network can be considered a graph in which the nodes are vertices and an edge exists between two vertices if they are close enough to communicate directly with each other. The graph formed by an ad hoc network is generally not planar (see Fig. 7, where the transmission range of each node contains all other nodes).

In order to construct a connected planar subgraph of the graph formed by the nodes in the ad hoc network, a well-known mechanism [32] is employed: an edge between two nodes A and B is included in the graph only if the intersection of the two circles with radii equal to the distance between A and B around those two nodes does not contain any other nodes. For example, in Fig. 7 the edge between A and C would not be included in the planar subgraph since B and D are contained in the intersection of the circles. It is important to realize that the decision as to whether an edge is within the planar subgraph can be made locally by each node, since each node knows the position of all its neighbors.

Based on the planar subgraph, a simple planar graph traversal is used to find a path toward the destination. The general concept is to forward the packet on faces of the planar subgraph progressively closer to the destination. Figure 8 from [31] shows how this traversal is carried out when a packet is forwarded from S toward D in recovery mode. On each face, the packet is forwarded along the interior of the face by using the right hand rule: forward the packet on the next edge counterclockwise from the edge on which it arrived. Whenever the line between source and destination intersects the edge along which a packet is about to be forwarded, check if this intersection is closer to the destination than any other intersection previously encountered. If this is true, switch to the new face bordering on the edge the packet was about to traverse. The packet is then forwarded on the next edge counterclockwise to the edge it was about to be forwarded along before switching faces. This algorithm guarantees that a path will be found from the source to the destination if there exists at least one such path in the original nonplanar graph.

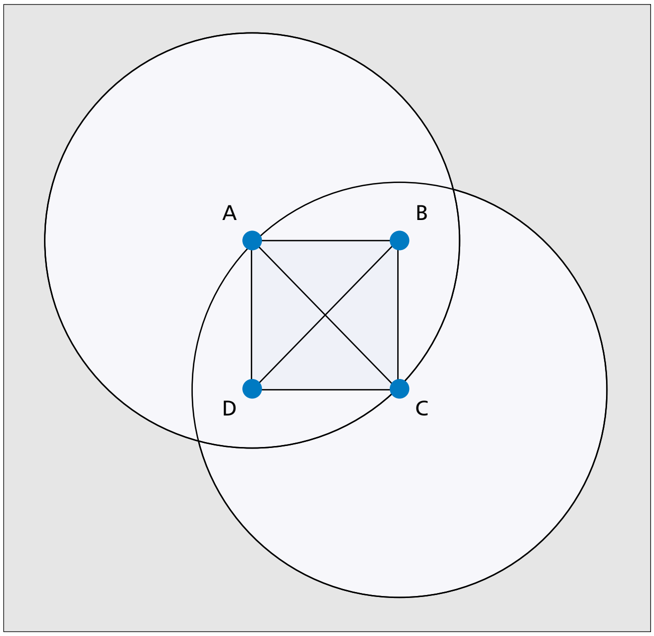
The header of a packet contains additional information such as the position of the node where it entered recovery mode, the position of the last intersection that caused a face change, and the first edge traversed on the current face. Therefore, each node can make all routing decisions based only on the information about its local neighbors. This includes the detection of an unreachable destination, when a packet traverses the first edge on the current face for the second time.

Restricted Directional Flooding

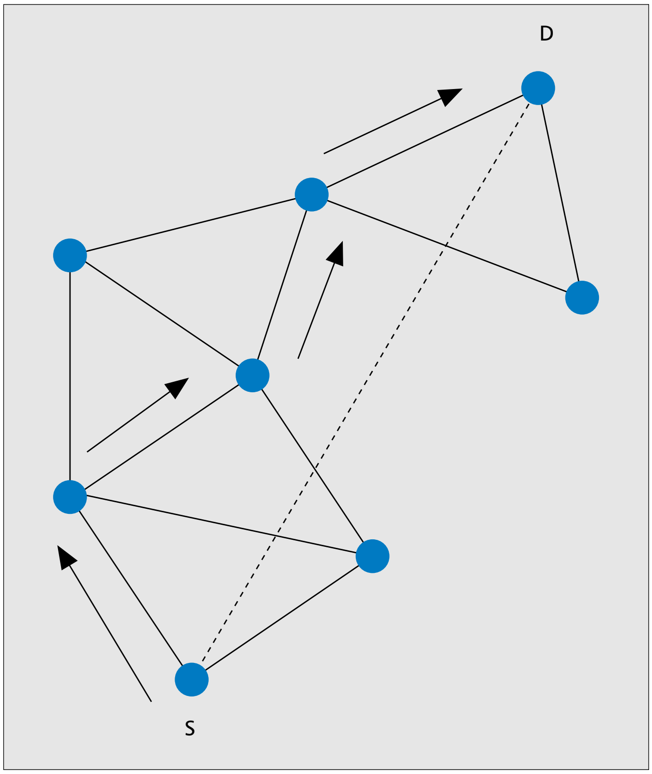
DREAM — In DREAM the sender S of a packet with destination D will forward the packet to all one-hop neighbors that lie "in the direction of D." In order to determine this direction, a node calculates the region that is likely to contain D, called the *expected region*. As depicted in Fig. 9, the expected region is a circle around the position of D as it is known to S. Since this position information may be outdated, the radius r of the expected region is set to $(t_1 - t_0)v_{max}$, where t_1 is the current time, t_0 is the timestamp of the position information S has about D, and v_{max} is the maximum speed that a node may travel in the ad hoc network. Given the expected region, the "direction toward D" for the example given in Fig. 9 is defined by the line between S and D and the angle φ. The neighboring hops repeat this procedure using their information on D's position. If a node does not have a one-hop neighbor in the required direction, a recovery procedure has to be started. This procedure is not part of the DREAM specification.

Location Aided Routing (LAR) — The Location Aided Routing proposal [33] does not define a location-based routing protocol but instead proposes the use of position information to enhance the route discovery phase of reactive ad hoc routing approaches. Reactive ad hoc routing protocols frequently use flooding as a means of route discovery. Under the assumption that nodes have information about other nodes' positions, this position information can be used by LAR to restrict the flooding to a certain area. This is done in a fashion similar to that of the DREAM approach.

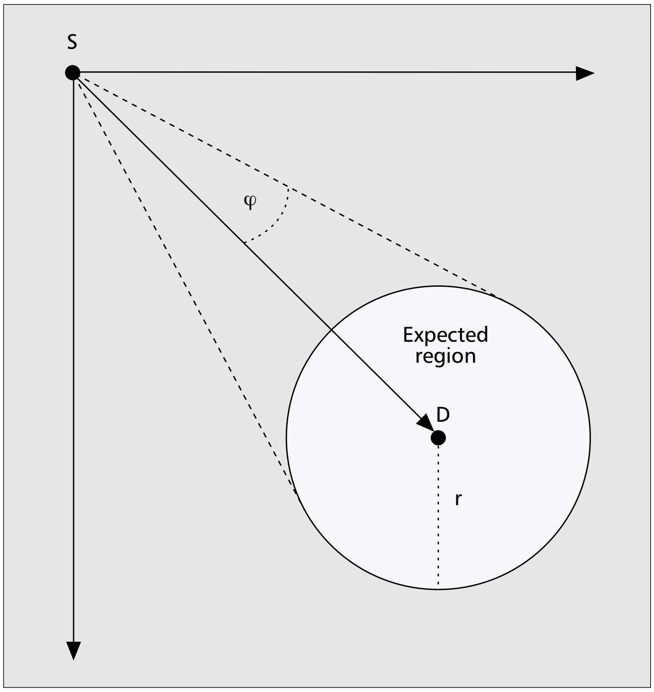
When node S wants to establish a route to node D, S com-



■ Figure 7. *Non-planar graph.*



■ Figure 8. *Planar graph traversal.*



■ Figure 9. *Example for the expected region in DREAM.*

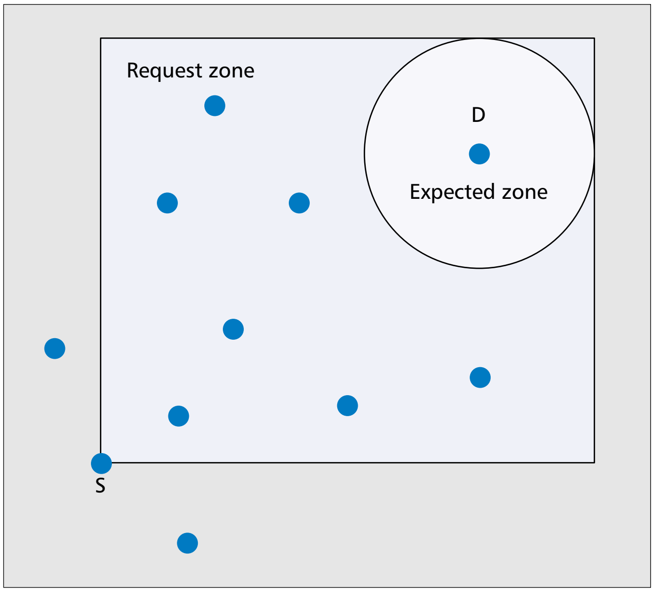
putes an *expected zone* for D based on available position information. If no such information is available LAR is reduced to simple flooding. If location information is available (e.g., from a route that was established earlier), a *request zone* is defined as the set of nodes that should forward the route discovery packet. The request zone typically includes the expected zone. Two request zone types have been proposed in [33]. The first is a rectangular geographic region. In this case, nodes will forward the route discovery packet only if they are within that specific region. This type of request zone is shown in Fig. 10. The second is defined by specifying (estimated) destination coordinates plus the distance to the destination. In this case, each forwarding node overwrites the distance field with its own current distance to the destination. A node is allowed to forward the packet again only if it is at most some δ (system parameter) farther away than the previous node.

Hierarchical Routing

In traditional networks, the complexity each node has to handle can be reduced tremendously by establishing some form of hierarchy. Hierarchical routing allows those networks to scale to a very large number of nodes. It is therefore a valid question to ask whether position-based routing for mobile ad hoc networks can also benefit from introduction of a hierarchy.

Terminodes Routing — One approach that combines hierarchical and position-based routing is part of the Terminodes project [34]. In Terminodes routing a two-level hierarchy is proposed [35]. Packets are routed according to a proactive distance vector scheme if the destination is close (in terms of hops) to the sending node. For long distance routing a greedy position-based approach is used. Once a long distance packet reaches the area close to the recipient, it continues to be forwarded by means of the local routing protocol. The authors of [35] show by means of simulations that the introduction of a hierarchy can significantly improve the ratio of successfully delivered packets and the routing overhead compared to reactive ad hoc routing algorithms.

In order to prevent greedy forwarding for long distance routing from encountering a local maximum, the sender includes a list of positions in the packet header. The packet must then traverse the areas at these positions on its way to the sender. The packet forwarding between the areas is done on a purely greedy basis. This approach can be thought of as position-based source routing. It requires that the sender know about appropriate positions leading to the destination. In Terminodes routing, the sender requests this information from nodes it is already in contact with (e.g., the nodes that are reachable using the local routing protocol). Once a sender has the information, it needs to check at regular intervals whether the path of positions is still valid or can be improved. Therefore, Terminodes long-distance routing contains elements of reactive ad hoc routing approaches.



■ Figure 10. *Example of request and expected zones in LAR.*

Criterion	DREAM	Quorum system	GLS	Homezone
Type	All-for-all	Some-for-some	All-for-some	All-for-some
Communication complexity (update)	$O(n)$	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(\sqrt{n})$
Communication complexity (lookup)	$O(c)$	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(\sqrt{n})$
Time complexity (update)	$O(\div n)$	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(\sqrt{n})$
Time complexity (lookup)	$O(c)$	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(\sqrt{n})$
State volume	$O(n)$	$O(c)$	$O(log(n))$	$O(c)$
Localized information	Yes	No	Yes	No
Robustness	High	Medium	Medium	Medium
Implementation complexity	Low	High	Medium	Low
Abbreviations: n = number of nodes, c = constant				

Table 1. *Characteristics of the presented location services.*

Grid Routing — A second method for position-based ad hoc routing containing hierarchical elements is proposed in the Grid [22] project. The location proxy technique described in [36] is similar to Terminodes routing: a proactive distance vector routing protocol is used at the local level, while position based routing is employed for long-distance packet forwarding. In Grid routing, however, the hierarchy is not only introduced to improve scalability, but also to allow nodes that do not know their own position to participate in the ad hoc network. The main idea is to have at least one position-aware node in each area where the proactive distance vector protocol is used. The position-aware nodes in this area may then be used as proxies: a position-unaware node uses the position of a position-aware node as its own position. Packets that are addressed to a position unaware node therefore arrive at a position-aware proxy and are then forwarded according to the information of the proactive distance vector protocol.

As a repair mechanism for greedy long-distance routing a mechanism called Intermediate Node Forwarding (INF) is proposed in [36]. As in Terminodes routing, the idea is to perform position-based source routing. If a forwarding node has no neighbor with forward progress, it discards the packet and sends a notification to the sender of the packet. The sender of the packet then chooses a single intermediate position randomly for a circle around the midpoint of the line between the sender and the receiver. Packets have to traverse that intermediate position. If the packet is discarded again, the radius of the circle is increased and another random position is chosen. This is repeated until the packets are delivered to the destination or until a predefined value has been reached and the sender assumes that the destination is unreachable.

Comparisons

In the following we compare the location services and forwarding strategies described in the previous sections. One key aspect of this comparison is how the individual approaches behave with an increasing number of nodes in the mobile ad hoc network. For the remainder of this section we assume that the density of nodes remains constant when the number of nodes increases. Therefore, the area covered by the ad hoc network increases as the number of nodes increases. Since the expected distance of two uniformly sampled points within a square of size $a \times a$ scales with a [37], it is expected that the number of hops between two uniformly sampled participants increases proportional to the square root of the increase in nodes.

Location Services

Table 1 shows the location services that have been discussed. The *type* indicates how many nodes participate in providing location information and for how many other nodes each of these nodes maintains location information. The *communication complexity* describes the average number of one-hop transmissions required to look up or update a node's position. The *time complexity* measures the average time it takes to perform a position update or a position lookup. The amount of state required in each node that maintains the position of other nodes is indicated by the *state volume*. Some location services provide *localized information* by maintaining a higher density or better quality of position information near the position of the node. This may be important if communication in an ad hoc network is mainly local. The *robustness* of a location service is considered to be low, medium, or high depending on whether it takes the failure of a single node, the failure of a small subset of all nodes, or the failure of all nodes to render the position of a given node inaccessible. The *implementation complexity* describes how well the location service is understood and how complex it is to implement and test it. This measure is highly subjective, and we explain our rating while discussing each location service.

DREAM is fundamentally different from the other position services in that it requires that all nodes maintain position information about every other node. The communication complexity of a position update and the position information maintained by each node scales with $O(n)$, while a position query requires only a local lookup, which is independent of the number of nodes. The time required to perform a position update in DREAM is a linear function of the diameter of the network, leading to a complexity of $O(\sqrt{n})$. Due to the communication complexity of position updates, DREAM is the least scalable position service and thus inappropriate for large-scale and general-purpose ad hoc networks. However, it also has interesting properties, making it suitable for specialized applications: it is very robust and provides localized information. Together with the packet forwarding proposed for DREAM, it is an interesting candidate for certain applications (e.g., local communication between cars in an emergency situation). The operation of DREAM is well understood in static and dynamic situations, and the protocol primitives can be realized in a straightforward fashion. Therefore, we assigned it a low implementation complexity.

The quorum system requires the same operations for position updates and position lookups. In both cases a constant number of nodes (the quorum) must be contacted. Each of these messages has a communication complexity and time complexity that depends linearly on the diameter of the network and thus scales with $O(\sqrt{n})$. The state information maintained in the backbone nodes is constant, since an individual backbone is formed for a fixed number of nodes. None of these figures includes the management of the virtual back-

Criterion	Greedy	DREAM	LAR	Terminodes	Grid
Type	Greedy	Restricted directional Flooding	Restricted directional Flooding	Hierarchical	Hierarchical
Communication complexity	$O(\sqrt{n})$	$O(n)$	$O(n)$	$O(\sqrt{n})$	$O(\sqrt{n})$
Tolerable position inaccuracy	Transmission range	Expected region	Expected region	Short-distance routing range	Short-distance routing range
Requires all-for-all location service	No	Yes	No	No	No
Robustness	Medium	High	High	Medium	Medium
Implementation complexity	Medium	Low	Low	High	High
Abbreviations: n = number of nodes					

Table 2. *Characteristics of the presented forwarding strategies.*

bone, which is not specified in [18]. The general robustness of the approach is medium, since the position of a node will become unavailable if a significant number of backbone nodes fail. However, this number of nodes is a parameter that can be freely configured for the position service. Furthermore, the position information is kept spatially distributed and independent. Therefore, the robustness seems to be higher than that of GLS or Homezone. A major drawback of the quorum system is its dependence on a non-position-based ad hoc routing protocol for the virtual backbone, which tremendously increases the implementation complexity and may compromise the scalability of this approach. However, the two position services, GLS and Homezone, can be thought of as specializations of the quorum system, eliminating this drawback.

GLS and Homezone are similar to each other in that each node selects a subset of all available nodes as position servers. For Homezone, position updates and lookups need to be sent to the virtual home region (VHR). The average distance from that region depends linearly on the diameter of the network; therefore, the communication/time complexity of Homezone is $O(\sqrt{n})$. The state information is constant, since each node should have a constant number of position servers in its Homezone. The performance of GLS is dependent on how the communication partners are distributed across the ad hoc network. If they are uniformly distributed, the number of position servers increases logarithmically with the number of nodes. Due to the localized strategy of forwarding updates and lookups, the communication and time complexity in this case is just a constant factor larger than in Homezone and remains at $O(\sqrt{n})$. The main trade-off between GLS and Homezone is in providing localized information and in the implementation complexity. GLS benefits greatly if the communication partners are close to each other and therefore outperforms Homezone for local communication. This is true since nodes in Homezone can be hashed to a distant VHR, leading to increased communication and time complexity, as well as problems if the VHR of a node cannot be reached. At the same time, the behavior of GLS in a dynamic environment and in the presence of node failures is more difficult to control than that of Homezone. Summarizing, both GLS and Homezone are very promising approaches for position services in general-purpose ad hoc networks.

Forwarding Strategies

Table 2 presents the forwarding strategies together with their evaluation criteria. The *type* describes the fundamental strategy used for packet forwarding. The *communication complexity* indicates the average number of one-hop transmissions required to send a packet from one node to another node under the assumption that the position of the destination is known. The forwarding strategies tolerate different degrees of inaccuracy with regard to the position of the receiver. This is reflected by the *tolerable position* inaccuracy criterion. Furthermore, the *requires all-for-all location service* criterion shows whether the forwarding strategy requires an all-for-all location service in order to work properly. The robustness of an approach is high if the failure of a single intermediate node does not prevent the packet from reaching its destination. It is medium if the failure of a single intermediate node might lead to the loss of the packet but does not require the setup of a new route. Finally, the robustness is low if the failure of an individual node might result in packet loss and the setup of a new route. By definition, the position-based strategies described in this article do not maintain routes and therefore have at least medium robustness. As for the location service, the *implementation complexity* describes how complex it is to implement and test a given forwarding strategy. This measure is highly subjective, and we explain our rating while discussing each forwarding strategy.

Greedy forwarding is both efficient, with a communication complexity of $O(\sqrt{n})$, and very well suited for use in ad hoc networks with a highly dynamic topology. The face-2 algorithm and the perimeter routing of GPSR are currently the most advanced recovery strategies. The only drawback of the current greedy approaches is that the position of the destination needs to be known with an accuracy of a one-hop transmission range; otherwise, the packets cannot be delivered. The robustness is medium since the failure of an individual node may cause the loss of a packet in transit, but it does not require setting up a new route, as would be the case in topology-based ad hoc routing. Due to the inclusion of a repair strategy like face-2 or perimeter routing, we consider the dynamic behavior and thereby the implementation effort to be of medium complexity.

The authors of GPSR have conducted a quantitative evaluation of the performance of their algorithms in a dynamic environment and compared it to Dynamic Source Routing (DSR) [6]. DSR is a reactive routing protocol for ad hoc networks that has been shown to be superior to many other existing reactive ad hoc routing protocols in [11]. The evaluation shows that GPSR performs better than DSR with regards to almost all criteria, including fraction of packets successfully delivered and routing protocol overhead. However, these simulations did not include the traffic and time required to look up the position of the destination; it was assumed that the position of the destination is accurately known by the sender.

Restricted directional flooding, as in DREAM and LAR has a communication complexity of $O(n)$ and therefore does not scale to large networks with a high volume of data transmissions. One difference between DREAM and LAR is that in DREAM it is expected that intermediate nodes update the position of the destination when they have better information than the sender of the packet. This is not done in LAR. The consequences are that DREAM packet forwarding requires and makes optimal use of an all-for-all location service, while LAR can work with any location service but does not benefit as much from an all-for-all location service if one is used. Both approaches are very robust against the failure of individual nodes and position inaccuracy, and very simple to implement. As mentioned above, this qualifies them for applications that require high reliability and fast message delivery for very infrequent data transmissions.

Terminodes and Grid routing both provide hierarchical approaches to position-based ad hoc routing. For long-distance routing both use a greedy approach and therefore have characteristics similar to those of greedy forwarding. However, due to the usage of a non-position-based approach at the local level, they are more tolerant of position inaccuracy on one hand, while being significantly more complex to implement on the other. Grid routing allows position-unaware nodes to use position-aware nodes as proxies in order to participate in the ad hoc network, while for Terminodes a GPS-free positioning service has been developed. The probabilistic repair strategy proposed by Grid is simpler and requires less state information than that of Terminodes. On the other hand, it may fail in cases where the Terminodes succeeds in finding a path from the sender to the destination.

Directions of Future Research

In the previous sections it has been shown that there are quite a number of different approaches to realizing location services and to performing position-based packet forwarding. However, there still exist a number of issues and problems that need to be addressed in future research. While we have provided a qualitative discussion of the current approaches, it is of great importance to also investigate them on a quantitative level. For non-position-based approaches such evaluations have been performed in [5, 11] with very interesting results. It can be expected that a quantitative comparison will yield more information on the strengths and weaknesses of the individual approaches, and potential improvements.

As discussed in the previous section, GLS and Homezone seem to be the most universally useful position services. Both are all-for-some approaches, and both make use of a hash function to identify the nodes that hold the position information about a given node. It will be a challenging task for future research to ensure that this hashing works properly in the face of very dynamic networks. Also, it is conceivable to develop all-for-some location services that do not use hashing. For example, one could use probabilistic methods, which may have better properties in very dynamic network environments.

There is one very important aspect of location services that is not considered by any existing approach: the problem of ensuring anonymity. When a persistent node identifier can be readily associated with its position, location privacy is hard to achieve. This is a major issue that needs to be addressed by future location services and forwarding strategies.

The strategies for position-based packet forwarding have been the subject of research since the mid-'60s. Greedy forwarding seems to be the method of choice for regular data, being both efficient and very well suited to ad hoc networks with a highly dynamic topology. Future research is likely to concentrate on two issues: the strategy employed to choose the next hop to which a packet is forwarded, and the repair mechanism used when greedy packet forwarding fails. The choice of the next hop depends decisively on the service provided by the wireless hardware. If it is possible to adapt the strength of the transmission signal to the distance between two communication partners, a strategy like NFP should be employed; otherwise, MFR seems to have an edge. While face-2 and perimeter routing are fairly advanced mechanisms, it could be worthwhile to investigate how to prevent very long "detours" of packets. Also, it would be interesting to gain a better understanding of how the recovery strategies behave when the topology of the network changes while a packet is being forwarded using the right hand rule. One additional topic that seems to be interesting is how to make greedy routing more tolerant of inaccurate position information.

As ad hoc networks become more common, it is very likely that connectivity among the individual ad hoc networks, as well as connectivity of any given ad hoc network and the global Internet will be desired. Most likely this will require the introduction of hierarchies, as done in the Terminodes and Grid projects. However, since the position of individual nodes in an ad hoc network will change much more frequently than the position of the ad hoc networks themselves, it could be argued that a hierarchical approach should use a location-based approach at the local level and topology-based routing over long distances and for Internet integration. It is also conceivable that a three-level hierarchy could be used. At the lowest layer a proactive routing protocol could be employed to aggregate a small number of nodes and increase robustness against positional errors. At the next layer a position-based approach might be used that scales well to ad hoc networks with numerous participants. Finally, the third layer would use proactive or reactive approaches to connect the ad hoc networks with each other and with the global Internet.

Summary

In this article we present a survey on position-based routing for mobile ad hoc networks. It is shown that the task of routing packets from a source to a destination can be separated into two distinct aspects:

- Discovering the position of the destination
- The actual forwarding of packets based on this information

We examined four location services for position discovery. In DREAM, the position information is flooded in the network. The time between flooding depends on the mobility of the node, while the range of each flooding is chosen so that nearby nodes are updated much more frequently than nodes farther away. Quorum-based position discovery requires identifying overlapping groups of participants. Updates are transmitted to one of those groups and position queries directed to another one. Since groups overlap, the required information is available in each group. The GLS approach works by hashing the ID of a node on the IDs of so-called location servers. These location servers are updated by the destination node with regard to its own position and queried by the source nodes that want to contact the destination node. Finally, the Homezone algorithms requires that the ID of a node be hashed on a position. All nodes close to this position are informed about the position of the node and provide this information to sources that want to contact it.

Forwarding packets based on position information was separated into three distinct areas. Greedy routing works by forwarding packets in the direction of the destination. If a local maximum is encountered, a repair strategy such as face-2 or GPSR's perimeter routing can be used to avoid dropping the packet. In restricted directional flooding, as used by DREAM and LAR, the packets are broadcasted in the general direction of the destination. On their way, the position information in the packets may be updated if a node has more current information about the destination's position. LAR differs from DREAM in that it uses the position information only to set up a route in an efficient manner. The actual data packets are routed with a position-independent protocol. In the Terminodes and Grid projects, routing is done hierarchically by means of a position-independent protocol at the local level and a greedy variant at the long-distance level.

We provide a qualitative evaluation of the presented approaches. Based on this evaluation, we argue that all-for-some location services, such as Homezone and GLS, in combination with greedy packet forwarding, are the most promising strategies for general-purpose position-based routing in mobile ad hoc networks. Approaches like DREAM and LAR could be used in situations where a small number of packets needs to be transmitted very reliably. Finally, we identify a number of research opportunities that could lead to further improvements, such as privacy and probabilistic methods for location services.

Acknowledgments

We would like to thank the anonymous reviewers for their detailed and helpful comments. Furthermore, Stefano Basagni, Ljubica Blazevic, Douglas S. J. De Couto, Jinyang Li, Ben Liang, and Ivan Stojmenovic gave us valuable feedback on the description and discussion of their algorithms.

References

[1] D. Beyer, M. D. Vestrich, and J. J. Garcia-Luna-Aceves, "The Rooftop Community Network: Free, High-Speed Network Access for Communities," Hurley and Keller, Edrs., *The First 100: New Options for Internet and Broadband Access*, MIT Press, 1999, pp. 75–91.

[2] C. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance-vector Routing (dsdv) for Mobile Computers," *Comp. Commun. Rev.*, Oct. 1994, pp. 234–44.

[3] P. Jacquet *et al.*, "Optimized Link State Routing Protocol. Internet draft, draft-ietf-manet-olsr-04.txt, work in progress, Sept. 2001.

[4] B. Bellur, R.Ogier, and F. Templin, "Topology Broadcast Based on Reverse-Path Forwarding (tbrpf)," Internet draft, draft-ietf-manet-tbrpf-01.txt, work in progress, Mar. 2001.

[5] S. R. Das, R. Castaneda, and J. Yan, "Simulation Based Performance Evaluation of Mobile, Ad Hoc Network Routing Protocols," *ACM/Baltzer MONET J.*, July 2000, pp. 179–89.

[6] D. Johnson and D. Maltz, *Mobile Computing, Chap. 5 — Dynamic Source Routing*, Kluwer Academic Publishers, 1996, pp. 153–81.

[7] V. Park and M. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. INFOCOM '97*.

[8] C. Perkins and E. Royer, "Ad-hoc on-demand Distance Vector Routing," *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. App.*, Feb. 1999, pp. 90–100.

[9] Z. Haas and M. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol," *ACM/IEEE Trans. Net.*, vol. 9, no. 4, Aug. 2001, pp. 427–38.

[10] E. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad Hoc Wireless Networks," *IEEE Pers. Commun.*, Apr. 1999, pp. 46–55.

[11] J. Broch *et al.*, "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols," *Proc. 4th ACM/IEEE Int'l. Conf. Mobile Computing and Networking MOBICOM '98*, Dallas, TX, USA, 1998, pp. 85–97.

[12] E. Kaplan, *Understanding GPS*, Artech House, 1996.

[13] S. Capkun, M. Hamdi, and J. Hubaux, "Gps-free Positioning in Mobile Ad Hoc Networks," *Proc. Hawaii Int'l. Conf. System Sciences*, Jan. 2001.

[14] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer*, vol. 34, no. 8, Aug. 2001, pp. 57–66.

[15] J. C. Navas and T. Imielinski, "Geographic Addressing and Routing," *Proc. 3rd ACM/IEEE Int. Conf. Mobile Comp. Net., MobiCom'97*, Sept. 1997.

[16] S. Basagni *et al.*, "A Distance Routing Effect Algorithm for Mobility (Dream)," *Proc. 4th Annual ACM/IEEE Int. Conf. Mobile Computing and Networking, MOBICOM '98*, Dallas, TX, USA, 1998, pp. 76–84.

[17] S. Basagni, I. Chlamtac, and V. Syrotiuk, "Geographic Messaging in Wireless Ad Hoc Networks," *Proc. 49th IEEE Int'l. Vehic. Tec. Conf.*, Houston, TX, USA, vol. 3, 1999, pp. 1957–61.

[18] Z. J. Haas and B. Liang, "Ad Hoc Mobility Management with Uniform Quorum Systems," *IEEE/ACM Trans. Net.*, vol. 7, no. 2, Apr. 1999, pp. 228–40.

[19] I. Stojmenovic, "A Routing Strategy and Quorum Based Location Update Scheme for Ad Hoc Wireless Networks," Tech. rep. TR-99-09, Comp. Sci., SITE, Univ. of Ottawa, 1999.

[20] J. Li *et al.*, "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. 6th Annual ACM/IEEE Int'l. Conf. Mobile Comp. Net.*, Boston, MA, 2000, pp. 120–30.

[21] R. Morris *et al.*, Carnet: A Scalable Ad Hoc Wireless Network System," *Proc. 9th ACM SIGOPS Euro. Wksp.: Beyond the PC: New Challenges for the Op. Sys.*, Kolding, Denmark, Sept. 2000.

[22] The grid project homepage, http://www.pdos.lcs.mit.edu/grid

[23] S. Giordano and M. Hamdi, "Mobility Management: The Virtual Home Region," Tech. report, Oct. 1999.

[24] I. Stojmenovic, "Home Agent Based Location Update and Destination Search Schemes in Ad Hoc Wireless Networks," Tech. rep. TR-99-10, Comp. Science, SITE, Unive. Ottawa, Sept. 1999.

[25] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Trans. Commun.*, vol. 32, no. 3, Mar. 1984, pp. 246–57.

[26] T.-C. Hou and V. O.K. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Trans. Commun.*, vol. 34, no. 1, Jan. 1986, pp. 38–44.

[27] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," *Proc. 11th Canadian Conf. Comp. Geo.*, Vancouver, Aug. 1999.

[28] R. Nelson and L. Kleinrock, "The Spatial Capacity of a Slotted Aloha Multihop Packet Radio Network with Capture," *IEEE Trans. Commun.*, vol. 32, no. 6, June 1984, pp. 684–94.

[29] P. Bose *et al.*, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. 3rd ACM Int'l. Wksp. Discrete Algorithms and Methods for Mobile Comp. and Commun.*, 1999, pp. 48–55.

[30] B. N. Karp, "Geographic Routing for Wireless Networks," Ph.D. thesis, Harvard Univ., 2000.

[31] B. Karp and H. T. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. 6th Annual ACM/IEEE Int'l. Conf. Mobile Comp. Net.*, Boston, MA, Aug. 2000, pp. 243–54.

[32] G. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," *Pattern Recognition*, vol. 12, no. 4, 1980, pp. 261–68.

[33] Y.-B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *ACM/Baltzer WINET J.*, vol. 6, no. 4, 2000, pp. 307–21.

[34] L. Blazevic *et al.*, "Self-Organization in Mobile Ad Hoc Networks: The Approach of Terminodes," *IEEE Commun. Mag.*, 2001.

[35] L. Blazevic, S. Giordano, and J. Le Boudec, "Self Organized Terminode Routing," Tech. rep. DSC/2000/040, Swiss Fed. Inst. Tech., Lausanne, Switzerland, 2000.

[36] D. S. J. De Couto and R. Morris, "Location Proxies and Intermediate Node Forwarding for Practical Geographic Forwarding," Tech. rep. MIT-LCS-TR824, MIT Lab. Comp. Sci., June 2001.

[37] B. Ghosh, "Random Distances within a Rectangle and Between Two Rectangles," *Bull. Calcutta Math. Soc.*, vol. 43, 1950, pp. 17–24.

Biographies

MARTIN MAUVE (mauve@informatik.uni-mannheim.de) received his M.S. and Ph.D. degrees in computer science from the University of Mannheim, Germany, in 1997 and 2000, respectively. He is currently an assistant professor at the University of Mannheim. His research interests include mobile communication, multimedia transport protocols, networked multimedia applications, and group communication.

JÖRG WIDMER (widmer@informatik.uni-mannheim.de) received his M.S. degree in computer science from the University of Mannheim, Germany, in 2000. He is currently employed at the University of Mannheim while pursuing his Ph.D. His research interests include multicast communication, congestion control, and ad hoc networking.

HANNES HARTENSTEIN (hannes.hartenstein@ccrle.nec.de) received his diploma degree in mathematics in 1995 and a Ph.D. degree in computer science in 1998, both from the University of Freiburg, Germany. He is currently a researcher with NEC Europe's Network Laboratory located in Heidelberg, Germany. His research interests include mobile communications, multimedia, and theoretical computer science.