Unicast Position-based Routing Protocols for Ad-Hoc Networks

Miss Laiha Mat Kiah

Department of Computer System and Technology
Faculty of Computer Science and Information Technology
University of Malaya
50603 Kuala Lumpur, Malaysia
misslaiha@um.edu.my

Liana Khamis Qabajeh

Faculty of Computer Science and Information Technology, University of Malaya, 58200 Kuala Lumpur, Malaysia
liana_tamimi@ppu.edu

Mohammad Moustafa Qabajeh

Department of Electrical and Computer Engineering, IIUM
58200 Kuala Lumpur, Malaysia
m_qabajeh@yahoo.com

Abstract: Wireless Ad-Hoc networks are collections of nodes that communicate without any fixed infrastructure. A critical problem in Ad-Hoc networks is finding an efficient and correct route between a source and a destination. The need for scalable and efficient protocols, along with the recent availability of small, inexpensive low-power positioning instruments justify adopting position-based routing protocols in mobile Ad-Hoc networks. This paper presents an overview of the existing Ad-Hoc routing protocols that make forwarding decisions based on the geographical position of a packet's destination. We also outline the main problems for this class of routing protocols. We conclude our findings by discussing opportunities for future research.

Keywords: position-based routing; location-aware routing; ad-hoc networks; wireless networks; routing protocols

1 Introduction

Ad-Hoc wireless networks are self-organizing multi-hop wireless networks, where all the nodes take part in the process of forwarding packets. Ad-Hoc networks can quickly and inexpensively be set up as needed since they do not require any fixed infrastructure, such as base stations or routers. Therefore, they are highly applicable in many fields such as emergency deployments and community networking.

The function of a routing protocol in Ad-Hoc network is to establish routes between different nodes. Ad-Hoc routing protocols are difficult to design in general. There are two main reasons for this: the highly dynamic nature of these networks due to the high mobility of the nodes, and the need to operate efficiently with limited resources, such as network bandwidth and the limited memory and battery power of the individual nodes in the network. Moreover, routing protocols in Ad-Hoc networks, unlike static networks, do not scale well due to frequently changing network topology, lack of predefined infrastructure like routers, peer-to-peer mode of communication and limited radio communication range [20].

For the aforementioned reasons, many routing protocols which are compatible with the characteristics of Ad-Hoc networks have been proposed. In general, they can be divided into two main categories: *topology-based* and *position-based*. *Topology-based* routing protocols use information about links that exist in the network to perform packet forwarding. In general, topology-based routing protocols are considered not to scale in networks with more than several hundred nodes [23].

In recent developments, *position-based* routing protocols exhibit better scalability, performance and robustness against frequent topological changes [20], [23]. Position-based routing protocols use the geographical position of nodes to make routing decisions, which results in improved efficiency and performance. These protocols require that a node be able to obtain its own geographical position and the geographical position of the destination. Generally, this information is obtained via *Global Positioning System (GPS)* and location services. 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. So the packets are delivered to the nodes in a given geographic region in a natural way. There are different kinds of position-based protocols which are categorized into three main groups: *restricted directional flooding*, *greedy* and *hierarchical routing protocols* [13] (to be discussed in *Section 2*).

This survey (which is an extended version of our work in [11]) gives an overview of a large percentage of existing unicast position-based routing protocols for mobile Ad-Hoc networks. We outlined the main problems which must be solved for this class of routing protocols and presented the solutions that are currently available. The discussed protocols are also compared with respect to the used location service, the used forwarding strategy, tolerability to position inaccuracy, robustness, implementation complexity, scalability, packet overhead, guaranteeing loop-freedom, probability of finding the shortest path and the suitable network density to be implemented in.

The protocols that have been selected for analysis are *MFR* [7], *DIR* [4], *GPSR* [1], *ARP* [20], *I-PBBLR* [23], *POSANT* [18], *DREAM* [15], *LAR* [24], *LARWB* [17], *MLAR* [19], *GRID* [21], *TERMINODES* [9] and *LABAR* [6]. It is worth nothing that many other position-based routing protocols exist for mobile Ad-Hoc networks; however, we have selected what we regard as representative for the existing approaches.

The rest of the paper is organized as follows. *Section 2* presents the basic idea and principles of position-based addressing and routing. *Section 3* gives an overview of the selected position-based routing protocols. *Section 4* outlines the differences between the discussed protocols. Directions for future research are discussed in *Section 5*. Finally, we conclude the paper in *Section 6*.

2 Basic Principles of Position-based Routing

The main prerequisite for position-based routing is that a sender can obtain the current position of the 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. The four possible combinations can be abbreviated as some-for-some, some-for-all, all-for-some and all-for-all [13].

There are three main packet-forwarding strategies used for position-based protocols: *greedy forwarding*, *restricted directional flooding* and *hierarchical* approaches. While their main objective is to utilize available position information in the Ad-Hoc routing, their means of achieving this are quite different. Most position-based protocols use *greedy forwarding* to route packets from a source to the destination. Greedy protocols do not establish and maintain paths from source to the destination; instead, a source node includes the approximate position of the recipient in the data packet and selects the next hop depending on the optimization criteria of the algorithm; the closest neighbor to the destination for example [13], [20]. Similarly, each intermediate node selects a next hop node until the packet reaches the destination. In order for the nodes to be able to do so, they periodically broadcast small packets (called beacons) to announce their position and enable other nodes to maintain a one-hop neighbor table. Such an approach is scalable and resilient to topology changes since it does not need routing discovery and maintenance; however, periodic beaconing creates a lot of congestion in the network and consumes the nodes' energy [23].

While the beaconing frequency can be adapted to the degree of mobility, a fundamental problem of inaccurate (outdated) position information is always present: a neighbor selected as a next hop may no longer be in transmission range. This leads to a significant decrease in the packet delivery rate with increasing node mobility. To reduce the inaccuracy of position information, it is possible to increase the beaconing frequency. However, this also increases the load on the network by creating a lot of congestion, increasing the probability of collision with data packets and the energy consumption of the nodes.

Unfortunately, greedy routing may not always find the optimal route, and it may even fail to find a path between source and destination when one exists [13]. An example of this problem is shown in Figure 1. Note that there is 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; therefore greedy forwarding will reach a local maximum from which it cannot recover. Generally, greedy forwarding works well in dense networks, but in sparse ones it fails due to voids (regions without nodes) [20].

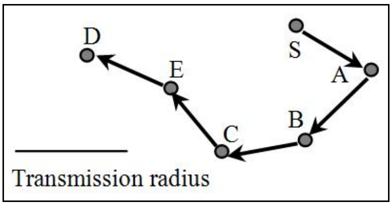

Figure 1
An example of greedy routing failure

In *restricted directional flooding,* the sender will broadcast the packet (whether the data or route request packet) to all single hop neighbors towards the destination. The node which receives the packet checks whether it is within the set of nodes that should forward the packet (according to the used criteria). If yes, it will retransmit the packet. Otherwise the packet will be dropped. In restricted directional flooding, instead of selecting a single node as the next hop, several nodes participate in forwarding the packet in order to increase the probability of finding the shortest path and be robust against the failure of individual nodes and position inaccuracy.

The third forwarding strategy is to form a *hierarchy* in order to scale to a large number of mobile nodes. Some strategies combine nodes' locations and hierarchical network structures by using zone-based routing such as *LABAR*. Others use dominating set routing such as *GRID*. Some others, such as *TERMINODES*, present a two-level hierarchy within them; if the destination is close to the sender, packets will be routed based on a proactive distance vector. Greedy routing is used in longer distances; therefore, these approaches have characteristics similar to those of greedy forwarding.

3 Overview of Selected Position-based Routing Protocols

In this section the selected protocols are described. For each protocol, we tried to summarize its main objectives, how it works and its advantages and disadvantages compared to other protocols. *Subsection 3.1* discusses greedy forwarding protocols, *Subsection 3.2* talks about restricted directional flooding ones and *Subsection 3.3* tackles hierarchical approaches. In the discussion part of each protocol, the following evaluation criteria have been taken into consideration:

- Location service type: indicates the type of the location service used with the given protocol; i.e., it shows how many nodes participate in providing location information and for how many other nodes each of these nodes maintains location information.
- Location service robustness: it is considered to be low, medium or high depending on whether the position of a given node will be inaccessible upon the failure of a single node, the failure of a small subset of the nodes or the failure of all nodes, respectively.
- Forwarding strategy type: describes the fundamental strategy used for packet forwarding.
- Forwarding strategy toleration to position inaccuracy: forwarding strategies tolerate different degrees of inaccuracy of the destination's position. This is reflected by the toleration to position inaccuracy criterion.
- Forwarding strategy robustness: the robustness of an approach is considered to be high if the failure (or absence due to mobility) 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 set up of a new route. Finally, robustness is low if the failure of an individual node might result in packet loss and the setting up of a new route. According to this definition, the routing protocols that begin data transmission immediately without the need for routing setup have at least medium robustness.
- Forwarding strategy implementation complexity: describes how complex it is to implement and test a given forwarding strategy. This measure is highly subjective and we will explain our opinion while discussing each protocol.
- Forwarding strategy scalability: describes the performance of the protocol with an increasing number of nodes in the network. It can be classified as follows: high scalability is used when a network grows as much as it needs and the approach is still able to maintain a good performance. Medium scalability means that an approach can handle networks with a reasonable size, but may have problems if it grows. Low scalability describes protocols which are

restricted to small networks. Since all the position-based routing protocols are scalable compared to topology-based ones, all the discussed protocols have at least medium scalability.

- Forwarding strategy packet overhead: refers to bandwidth consumption due to a higher number of signaling packets. The packets' sizes were not taken into consideration since all the discussed protocols are considered to have small packets, compared to secure protocols for example. Note that position-based routing protocols have lower packet overhead compared to topology-based ones. Hence all the discussed protocols have at most medium packet overhead.
- Loop-freedom: any routing protocol should be inherently loop-free to preserve the network resources and guarantee the correct operation of the protocol. Therefore, the discussed protocols are classified as having or not having loop-freedom property.
- Optimal path: this is used to indicate the probability that the protocol will find and use the shortest path for data packet relay.
- Density: indicates whether the protocol is more suitable to be implemented in dense or/and sparse networks.

3.1 Greedy Forwarding Protocols

This section discusses selected greedy forwarding routing protocols. The discussed protocols are *MFR* [7], *DIR* [4], *GPSR* [1], *ARP* [20], *I-PBBLR* [23] and *POSANT* [18].

3.1.1 MFR

Some greedy position-based routing protocols, as *Most Forward within distance R (MFR)* [7], try to minimize the number of hops by selecting the node with the largest progress from the neighbors, where progress is defined as the projection of the distance of the next hop from the sender on the straight line between the sender and the destination [7]. In Figure 2, if *MFR* is used, the source *S* will choose node *A* as the next hop since it has the largest progress to the destination *D*.

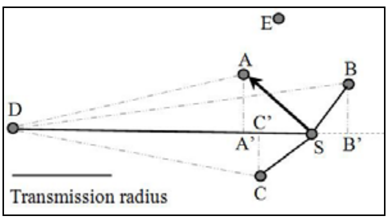

Figure 2
MFR example

MFR has the shortcomings of either not guaranteeing to find a path to the destination or finding a path which is much longer than the shortest path. Moreover, nodes periodically should broadcast beacons to announce their positions and enable other nodes to maintain a one-hop neighbor table.

MFR is the only progress-based algorithm competitive in terms of hop count [16]. However, choosing the node with the largest progress as the next hop will increase the probability that the two nodes will disconnect from each other before the packet reaches the next hop. So the packet drop rate increases greatly, especially in highly mobile environments. Such a situation is very common due to neighbor table inconsistency [20].

Discussion

As other greedy forwarding protocols, all nodes in *MFR* maintain a one-hop neighbor table; i.e., *MFR* uses all-for-some location service. Hence, a given node will be inaccessible upon the failure of a subset of the nodes; its location service has medium robustness. However, the technique used to enable the source knows the position of the destination is not discussed. Greedy forwarding is both efficient and very well suited for use in Ad-Hoc networks with a highly dynamic topology [13]. However, one important drawback of 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 [13].

MFR 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. Such an approach is very easy to implement and scalable since it does not need routing discovery and maintenance [22]. Moreover, it has a low packet overhead due to its small number of small-size packets.

MFR is probed to be a loop-free algorithm [8] since it always forces a message to make a step closer to the destination. Generally, greedy routing may not always find the optimum route and it may even fail to find a path between the source and destination when one exists [22]; the probability of finding the optimal path is considered as medium. Finally, all basic distance, progress and direction based methods such as *MFR* and *DIR* have high delivery rates in dense graphs, and low delivery rates in sparse ones [16].

3.1.2 DIR

Compass routing algorithms, such as *DIR* [4], try to minimize the spatial distance that a packet travels and are based on forwarding the packet to the neighboring node that minimizes the angle between itself, the previous node and the destination [13]. The source or intermediate node *A* uses the location information of the destination *D* to calculate its direction. Then the message *m* is forwarded to the neighbor *C*, such that the direction *AC* is closest to the direction *AD*. This process is repeated until the destination is, eventually, reached [16]. Consider the network in Figure 3, where the transmission radius is as indicated in the figure. The direction *AC* is closest to direction *AD* among candidate directions *AS*, *AB*, *AC*, *AG* and *AF*. So the path selected by *DIR* method is *SACD*.

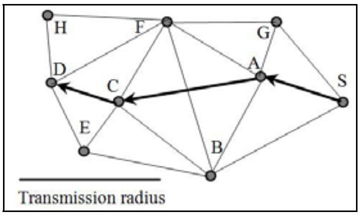

Figure 3
DIR example

As a greedy protocol, *DIR* suffers from congestion created by frequent beaconing, and it may not always find the optimum route, and it may even fail to find a path between source and destination if it exists.

The *DIR* method, and any other method that includes forwarding the message to the neighbor with closest direction, such as *DREAM* [15], are not loop-free, as is shown in [8] using the counterexample in Figure 4. The loop consists of four nodes denoted *S*, *B*, *C* and *A*. The transmission radius is as indicated in the figure. Let the source be any node in the loop, e.g. *S*. Node *S* selects node *B* to forward the message, because the direction of *B* is closer to destination *D* than the direction of its other neighbor *A*. Similarly, node *B* selects *C*, node *C* selects *A* and node *A* selects *S*.

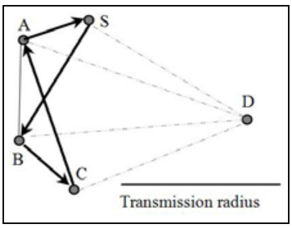

Figure 4
A loop in the directional routing

Discussion

As a greedy forwarding protocol, *DIR* has the same criteria as *MFR* except that *DIR* and any other method that includes forwarding the message to the neighbor with closest direction, such as *DREAM*, are not loop-free.

3.1.3 GPSR

Nearly Stateless Routing with Guaranteed Delivery are schemes where nodes maintain only some local information to perform routing. The face routing and *Greedy-Face-Greedy (GFG)* schemes were described in [14]. In order to ensure message delivery, the face routing (called *perimeter* algorithm in [1]) constructs a planar and connected so-called Gabriel subgraph of the unit graph, and then applies routing along the faces of the subgraph (e.g. by using the right hand rule) that intersect the line between the source and the destination. If a face is traversed using the right hand rule then a loop will be created; since a face will never exist. Forwarding in the right hand rule is performed using the directional approach. To improve the efficiency of the algorithm in terms of routing performance, face routing can be combined with algorithms that usually find shorter routes, such as the greedy algorithm to yield *GFG* algorithm [14]. Routing is mainly greedy, but if a mobile host fails to find a neighbor closer than itself to the destination, it switches the message from ‘greedy’ state to ‘face’ state [13].

Authors in [1] transformed *GFG* algorithm into *Greedy Perimeter Stateless Routing (GPSR)* protocol by including IEEE 802.11 medium access control scheme. The perimeter routing strategy of the *GPSR* is based on planar graph traversal and is proposed to address the local maximum problem of greedy forwarding [8]. It is performed on a per-packet basis and does not require the 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 [13]. *GPSR* guarantees that a path will be found from the source to the destination if there exists at least one such path in the original non-planar graph [13].

In [1] *GPSR* was experimented and compared with the non-position based protocol, *Dynamic Source Routing (DSR)* [3]. *GPSR* protocol consistently delivered over 94% of the data packets successfully; it is competitive with *DSR* in 50 node networks, and increasingly more successful than *DSR* as the number of nodes increases. The routing protocol traffic generated by *GPSR* was constant as mobility increased, while *DSR* must query longer routes with longer diameter and do so more often as mobility increases. Thus, *DSR* generates drastically more routing protocol traffic in simulations with over 100 nodes [1]. Therefore, the scalability seems to be the major advantage of this class of algorithms over source-based protocols. However, these simulations did not include the traffic and the time required to look up the position of the destination. It was also assumed that the position of the destination is accurately known by the sender [13].

Nearly stateless schemes are likely to fail if there is some instability in the transmission ranges of the mobile hosts, when the network graph includes nodes with irregular transmission ranges [18]. Transmission range instability means that the area a mobile host can reach is not necessarily a disk. This unstable situation occurs if there are obstacles (e.g. buildings, bad weather) that disrupt the radio transmission [16]. In *GPSR*, as with other greedy forwarding protocols, periodic beaconing creates lot of congestion in the network and consumes nodes' energy. In addition, *GPSR* uses link-layer feedback from *Media Access Control (MAC)* layer to route packets; such feedbacks are not available in most of the *MAC* layer protocols [20]. Finally, planarizing the underlying graph is computationally expensive and requires up-to-date neighborhood information [20].

Discussion

GPSR exhibits all the properties of greedy forwarding except that its implementation effort is considered to be of medium complexity due to planarizing underlying network and using perimeter routing. On the other hand, using the right hand thumb rule and perimeter mode routing made it applicable in sparse networks as well as dense ones.

3.1.4 ARP

Another scalable position-based routing protocol is *Angular Routing protocol (ARP)* [20]. In *ARP*, nodes emit a hello packet on a need-basis (non-periodic) at a rate proportional to their speeds. These hello packets enable each node to maintain a one hop neighbor table. *ARP* uses geographic forwarding to route packets to the destination. If geographic forwarding fails, an angle-based forwarding scheme is used to circumvent voids in sparse networks. *ARP* does not need any link-layer feedbacks like *GPSR*. If a source wants to send a packet to a specific destination, it selects as the next hop the node among its neighbors geographically closest to the destination. Each intermediate node follows this next hop selection criterion. Thus, at each hop the packet progresses towards the destination by a distance $\leq 0.9\ R$, where R is the radio range of the node. This is done to avoid the problem of leaving the next hop node out from the transmission range of the current node.

If no node is closer to the destination than the source node, or any intermediate node, then the node selects a neighboring node that creates a minimum angle, among available neighbors. Figure 5 shows the angle-based forwarding to circumvent voids. The intermediate node *B* has no neighbors closer to the destination *D* than itself. In such a situation *B* selects a next hop that forms a minimum angle towards destination; i.e., node *C*.

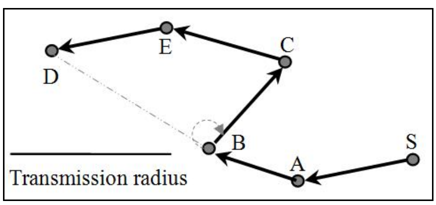

Figure 5

Angle-based forwarding to circumvent voids in *ARP*

After selecting a next hop node, the intermediate node appends its ID to the packet header. For each data packet, the *ARP* header memorizes a maximum of *k* last visited hops in order to avoid selecting a next-hop whose ID presents in the *ARP* header. This memorization technique helps *ARP* avoid local loops but does not guarantee its loop freedom. It is clear that assigning *k* a small value will decrease the percentage of local loops avoided. On the other hand, assigning it a large value will enlarge the size of the packet, which in turn will increase the packet overhead.

The simulations in [20] showed that *ARP* is scalable and achieves a high packet delivery rate while incurring low overhead compared to *GPSR*. Emitting hello packets on a need-basis reduces the problems associated with beaconing. Also, using the angle-based forwarding to circumvent voids increases the probability of finding a path (not necessarily the optimal one) in sparse networks.

Discussion

ARP exhibits all the properties of greedy forwarding except that by memorizing the last visited hops in the packet header it avoids local loops; however, this does not guarantee its loop freedom. Moreover, its use of an angle-based forwarding scheme to circumvent voids makes it applicable in sparse networks as well as dense ones.

3.1.5 I-PBBLR

Most position-based routing protocols use forwarding strategies based on distance, progress or direction. *Improved Progress Position Based BeaconLess Routing algorithm (I-PBBLR)* [23] combines the traditional progress with the direction metric to form the improved progress definition. There are many methods to combine the progress with direction, such as weighted addition and simple multiplication. The authors have chosen the cosine of the angle since its value is between 0 and 1, and it is even. If the traditional progress is multiplied by the cosine of the angle, both the minimum and maximum of the progress are not changed. It also satisfies the need that the node with a smaller angle will forward packet earlier. Finally, they guarantee loop freedom as the packets are always forwarded a step closer to the destination.

I-PBBLR tries to eliminate the beaconing drawbacks by using a beaconless protocol. In beaconless protocols, the sender makes non-deterministic routing decisions allowing opportune receiving nodes to determine a packet's next-hop through contention at transmission time. In *I-PBBLR*, if a source node has a data packet to send, it first determines the position of the destination, stores these geographical coordinates along with its own current position in the header of the packet, and broadcasts the packet to all neighboring nodes (since it does not possess knowledge of neighboring nodes' positions).

Nodes located within the forwarding area of the relaying node apply *Dynamic Forwarding Delay (DFD)* prior to relaying the packet, whereas nodes outside this area drop the received packet. The value of the *DFD* depends on the relative position coordinates of the current, previous and destination nodes. Eventually, the node that computes the shortest *DFD* forwards the packet first by broadcasting it to all neighboring nodes after replacing the previous node's position in the header with its own current position. Every node in the forwarding area detects the further relaying of the packet and cancels its scheduled transmission of the same packet. This mechanism allows selecting one neighbor as the next hop in a completely distributed manner, without having knowledge of the neighboring nodes, which is achieved by applying the concept of *DFD*. The simulation results showed that position-based beaconless routing using the improved progress reduced the overhead and increased the delivery rate by 3-5% compared with using the traditional progress.

Discussion

I-PBBLR inherited all the properties of greedy forwarding; however, the used location service was not discussed at all. Moreover, using a beaconless protocol slightly increases the robustness and scalability, reduces the packet overhead, improves the performance in sparse networks and increases tolerability to position inaccuracy compared to traditional greedy protocols. Finally, using the improved progress guarantees loop freedom as the packet is always forwarded a step towards the destination.

3.1.6 POSANT

Some position-based routing algorithms, such as *GPSR*, fail to find a route from a source to a destination (or they find a route that is much longer than the shortest path) when the network contains nodes with irregular transmission ranges. On the other hand, routing algorithms based on *Ant Colony Optimization (ACO)* guarantee message delivery and converge on a route which is very close to the optimal route even if the network contains nodes with different transmission ranges. However, *ACO* algorithms use a large number of messages and need a long time before the routes are established. *POSition-based ANT colony routing Algorithm for mobile Ad-Hoc networks (POSANT)* [18] is a reactive routing algorithm which is based on *ACO* and uses information about the location of nodes in order to reduce the route establishment time while keeping the number of generated ants smaller in comparison to other ant-colony-based routing algorithms.

In *POSANT*, to establish a route from a source node *S* to a destination node *D*, neighbors of *S* are partitioned into 3 zones as shown in Figure 6. After that, *S* launches *n* forward ants with unique sequence numbers from each zone at regular time intervals. *POSANT* assumes that each node maintains a table of the values of pheromone trails assigned to its outgoing links for different destinations. Upon receiving a packet for a specific destination, a node will check if there is at least one pheromone trail for that destination; this pheromone trail will be used for making a stochastic decision to select the next hop. If no such pheromone trail exists, a pheromone trail is initialized to each outgoing link. The amount of the deposited pheromone on each link depends on the zone of the corresponding neighbor. The motivation is that in most cases a shortest route passes through the nodes which are closer to the direction of the destination.

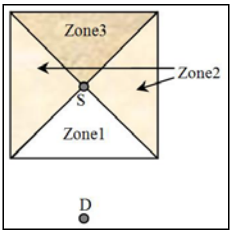

Figure 6
Different zones of *S* for destination node *D*

Whenever a forward ant enters a node from one of its neighbors, the identifier of the neighbor, the sequence number of the ant and the identifier of the destination will be stored. Repeated forward ants will be destroyed. When a forward ant reaches the destination, it is destroyed and a backward ant with the same sequence number is sent back to the source. Moving from node *B* to node *A*, the backward ant increases the amount of pheromone stored in edge *AB*. An evaporation process causes the amount of pheromone deposited in each link to decrease with time.

The above stochastic strategy establishes multiple paths between the source and destination. As a result, *POSANT* is a multipath routing algorithm. Multipath routing reduces the chance of congestion in the network; on the other hand, they can lead to out-of-order packet delivery problems.

Consider if a node *A* realizes that the link to *B* is broken and there is a pheromone trail corresponding to link *AB* for *D* in the pheromone table of *A*. In this case the stochastic data routing will continue, but if there is no pheromone trail for *D* in any of the other outgoing links of *A*, *A* sends a message to its neighbors to inform them that there is no route to *D* from *A*. Upon receiving this message, these neighbors do the same as if the link to *A* is broken. If the only outgoing link of the source node that has a pheromone trail for *D* breaks or a message from this link is received that states there is no route to *D*, a new route establishment process will begin and sending data packets will be suspended until a new route is found. Simulations in [18] showed that *POSANT* has a shorter route establishment time while using a smaller number of control messages than other ant colony routing algorithms.

Discussion

In *POSANT*, the used location service was not discussed. However, the used forwarding strategy is multiple greedy forwarding with the pheromone trail value used as the optimization criteria; a source launches many forward ants from different zones at regular time intervals and the pheromone trails' values are used for making a stochastic decision to select the next hop.

POSANT is tolerant of position inaccuracy due to the forwarding ants being sent to different zones (not to specific nodes' positions) and due to its use of the pheromone trail value as the optimization criterion (which does not depend on the exact position of nodes).

POSANT's robustness is considered to be medium, since the failure of a single node might result in packet loss but does not result in a new route establishment, except if the only outgoing link of the source node that has a pheromone trail for *D* breaks or a message from this link is received stating that there is no route to *D*.

The use of periodic multiple greedy forwarding caused *POSANT*'s implementation complexity, scalability and packet overhead to be considered as medium. *POSANT* is guaranteed to be loop-free since repeated forward ants are destroyed. Moreover, it has a high probability of finding the optimal path since it is based on *ACO*, which guarantees message delivery and converges to a route which is very close to the optimal route, even if the network contains nodes with different transmission ranges.

Finally, *POSANT* may be implemented in both dense and sparse networks. *POSANT* is better for sparse networks than traditional greedy forwarding because if no pheromone trail exists, the route discovery packet will not be dropped; however, pheromone trail initialization is done. Moreover, if it is used in dense networks it will have good performance due to low processing and medium packet overheads.

3.2 Restricted Directional Flooding

This section discusses a selected set of existing restricted directional flooding routing protocols. The selected protocols are *DREAM* [15], *LAR* [24], *LARWB* [17] and *MLAR* [19].

3.2.1 DREAM

Distance Routing Effect Algorithm for Mobility (DREAM) [15] is an example of restricted directional flooding routing protocols, within which the sender will broadcast the packet towards nodes in a limited sector of the network, that is, to all single hop neighbors towards the destination. *DREAM* algorithm is a proactive protocol that uses a limited flooding of location update messages [16]. In *DREAM*, each node maintains a position database that stores position information about all other nodes in the network. Its location service can therefore be classified as an all-for-all approach. Thus, each node regularly floods packets to update the position information maintained by the other nodes. The higher the speed of a node, the more the frequency at which it sends position updates. Also, the distance that a position update may travel before it is discarded provides accurate position information in the direct neighborhood of a node and less accurate information at nodes farther away, but this does not cause a problem since intermediate hops are able to update the position information contained in the data packet [7], [11]. In *DREAM* the message is forwarded to all neighbors whose direction belongs to the region that is likely to contain the destination *D*, called the expected region. The expected region is determined by the tangents from the source *S* to the circle centered at *D* and with radius equal to a maximal possible movement of *D* since the last location update [8]. The neighboring hops repeat this procedure using their information on *D*'s position.

Figure 7 is an example of expected region in *DREAM*. If a node does not have a neighbor in the required direction, a recovery procedure must be started. However, this procedure is not part of *DREAM* specification [13].

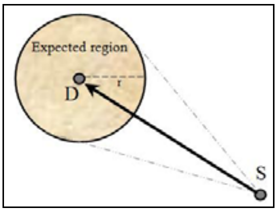

Figure 7
Example of the expected region in *DREAM*

Since *DREAM* uses restricted directional flooding to forward data packets themselves, there will be multiple copies of each packet at the same time. This increases the probability of using the optimal path; however, it decreases its scalability to large networks with a high volume of data transmissions and makes it more suitable for applications that require a high reliability and fast message delivery for infrequent data transmissions.

Discussion

DREAM is robust against position inaccuracy since it uses the expected region concept. It has higher communication complexity than greedy ones and therefore has less scalability to large networks; its scalability and packet overhead are considered to be medium. Moreover, it forwards packets to neighbors with closest direction, so it is not loop-free [8]. On the other hand, it can very simple be implemented and has high probability to find the optimal path. Finally it may be implemented in both dense and sparse networks; it is better for sparse networks than greedy forwarding, and even if it is used in dense ones it will have good performance due to low processing and medium packet overheads.

DREAM's location service is fundamentally different from other location services in that it requires that all nodes maintain position information about every other node. This leads to large overhead due to the position updates and large position information maintained by each node. Hence, *DREAM*'s location service is the least scalable position service and thus not appropriate for large-scale and general-purpose Ad-Hoc networks. On the other hand, a position query requires only a local lookup and the position of a given node will be inaccessible only upon the failure of all nodes, which makes it very robust.

DREAM is very robust against the failure of individual nodes since the data packet goes through multiple paths, so the failure of a single intermediate node does not prevent the packet from reaching its destination. This qualifies it for applications that require a high reliability and fast message delivery for very infrequent data transmissions [13].

3.2.2 LAR

Like *DREAM*, *Location-Aided Routing (LAR)* [24] is an example of restricted directional flooding routing protocols; however, partial flooding is used in *LAR* for path discovery purpose and in *DREAM* for packet forwarding [16]. Thus, *LAR* 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 [13]. If no information is available in the source about the position of the destination, *LAR* is reduced to simple flooding [13]. Otherwise, the expected zone (the area containing the circle and two tangents) is fixed from the source and defined based on the available position information (e.g., from a route that was established earlier) [7], [24]. 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 schemes have been proposed in [24]. The first scheme 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 Figure 8.

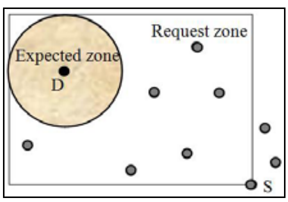

Figure 8
Example of request and expected zones in scheme 1 of *LAR*

In *LAR* scheme 2, the source or an intermediate node will forward the message to all nodes that are closer to the destination than itself. Thus, the node that receives the route request message will check whether it is closer to the destination than the previous hop, and if so it will retransmit the route request message; otherwise, it will drop the message.

In order to find the shortest path in the network level, instead of selecting a single node as the next hop, several nodes will be selected for managing the route request message, and each of them will put its *IP* address in the header of the request packet. Therefore, the route through which the route request message is passed will be saved in the header of the message [17]; the message size will grow as it goes far from the source and the routing overhead will increase. In *LAR*, if the discovered route breaks for any reason, the route discovery process must start again.

Discussion

As a restricted directional flooding protocol, *LAR* exhibits some of *DREAM*'s properties, such as robustness against position inaccuracy, high communication complexity, medium scalability and packet overhead, not guaranteeing loop-freedom, implementation simplicity, high probability to find the optimal path and suitability for implementation in both dense and sparse networks.

On the other hand *LAR* does not require all nodes to maintain position information about every other node, as in *DREAM*. Instead, it simply uses the available position information from a route that was established earlier.

LAR is robust during route discovery since the route discovery packet goes through multiple paths; however, after route setup, it is like any other protocol that depends on route setup before sending the data packets; i.e., the failure of a single node might result in packet loss and the setting up of a new route. Hence, its robustness is considered to be low. On the other hand, its establishing of a route before beginning data sending makes it more suitable than *DREAM* in cases that require high volumes of data transmissions.

3.2.3 LARWB

Routes in *LAR* are often broken due to mobility [19]. New routes must be rediscovered in order to continue the routing of packets in the queue. This problem was solved by *Location-Aided Routing With Backup (LARWB)* [17] since another route is selected as a backup route which is used when a breakage appears in the primary route. Selecting an appropriate backup route can be done by considering two points: the primary and the backup routes must have the minimum common nodes; and the backup route should have a low probability of having nodes that may leave the radio range of their previous hop node.

Experimental results in [17] show that by using *LARWB*, the number of nodes which participate in routing operation, the average number of exchanged messages in route discovery process and also the average time of route discovery were considerably reduced.

Discussion

LARWB exhibits all the properties of *LAR* except that its robustness is considered to be medium since the failure of a single node might result in packet loss but does not result in setting up of a new route due to the usage of the route backup. This route backup also reduces the number of routing packets; however, we still cannot consider *LARWB*'s packet overhead as low as that in greedy.

3.2.4 *MLAR*

Multipath Location Aided Routing (MLAR) [19] is a multipath routing version of *LAR* that works efficiently in both 2-Dimensional (2D) and 3-Dimensional (3D) networks. Here multipath means the caching of alternate paths to be used in the event of the failure of the primary path and not the use of multiple simultaneous paths, which can lead to out of order packet delivery problems. The two most recently received routes are cached even if they are longer. It is believed that the most recently received path (even if it is longer) is the path most likely to succeed since mobility is more likely to break an older path. However, a routing protocol with longer average hop counts may have lower packet delivery rate. This is because the probability of a packet being dropped is higher if packets traverse longer paths. If the second path also fails a new route request cycle is initiated. Since the packet header contains the entire source route, all paths are checked easily as being loop free at each node that stores routes.

In order to be able to compare *MLAR* to other existing protocols, the authors have extended ns-2 to support 3D mobility models and routing protocols. The simulation results demonstrated the performance benefits of their multipath position based algorithm over a multipath non position based algorithm, *Ad-Hoc On-demand Multipath Distance Vector routing (AOMDV)* [12], as well as with both their single path versions (*LAR* and *Ad-Hoc On-demand Distance Vector routing (AODV)* [2]) in both 2D and 3D. Only *AOMDV* consistently performs better than *MLAR* in terms of overall packet delivery, but this was at the cost of more frequent flooding of control packets and so more bandwidth. Thus, *MLAR* has lower bandwidth and energy usage than non position-based protocols and is more scalable and efficient. Moreover, *MLAR* performs consistently better than *LAR* in terms of packet delivery ratio, by as much as 30% in some cases.

Discussion

MLAR has similar criteria as that of *LAR* except that its robustness is medium since the failure of a single node might result in packet loss but does not result in setting up of a new route due to the usage of the alternate paths. These alternate paths also reduce the packet overhead; however, it is still higher than that of greedy.

Since *MLAR* caches the most recently received routes, the probability of using the optimal path is very low. Lastly, since the packet header contains the entire source route, all paths are checked easily as being loop free at each node that stores routes; loop freedom is guaranteed.

3.3 Hierarchical Routing Protocols

This section considers some hierarchical routing protocols, namely *GRID* [21], *TERMINODES* [9] and *LABAR* [6].

3.3.1 GRID

The two main strategies used to combine nodes location and hierarchical network structures are the zone-based routing and the dominating set routing [16]. In *GRID* algorithm [21] the dominating set concept is applied. A set is dominating if all the nodes in the system are either in the set or neighbors of nodes in the set. Routing based on a connected dominating set is a promising approach, since the searching space for a route is reduced to nodes in the set. *GRID* tries to exploit location information in route discovery, packet relay and route maintenance. In *GRID* the geographic area is partitioned into a number of squares called grids. In each grid, one mobile host (the one nearest to the physical center of the grid) will be elected as the leader of the grid. The size of each grid depends on transmission radius *R*, and several options are proposed, with the general idea of one leader being able to communicate directly with leaders in neighboring grids, and all nodes within each grid being connected to their leaders. Routing is then performed in a grid-by-grid manner through grids' leaders, and non-leaders have no such responsibility. Hence, the number of packets related to route search is insensitive to the network density. In fact, the cost slightly goes down as the host density increases, since routes become more stable with denser hosts.

In *GRID*, efforts are made in two directions to reduce the route search cost: using the locations of source and destination to confine the search range (like request zone in *LAR*) and delegating the searching responsibility to the gateway hosts. One attractive feature of *GRID* is its strong route maintenance capability since when a leader moves, another leader from the same grid replaces it through a handoff procedure. The probability of route breakage due to a node's roaming is reduced since the next hop is identified by its physical location, instead of by its address. *GRID* uses a specific field to detect duplicate request packets from the same source, so endless flooding of the same request can be avoided; i.e., it is loop free routing.

Simulations in [21] show that *GRID* can reduce the probability of route breakage, reduce the number of route discovery packets and lengthen routes' lifetimes. On the other hand, simulations also show that *GRID* uses longer paths than those used with *LAR*, since the former always confines relay hosts to gateway hosts while *LAR* tries to search the route with the smallest host count. Also, the authors do not elaborate on the route maintenance required if a grid remains empty after its leader and only node leaves it.

Feeney and Nillson in [10] and Shih et al. in [5] concluded that the node power consumption when idle is nearly as large as when receiving data. Also, a node in idle mode spends about 15-30 times more energy than if it is in sleep mode. Therefore, the development of protocols that have as many nodes as possible sleeping, such as *GRID*, will significantly save network energy.

Discussion

GRID is a hierarchical routing that applies the concept of dominating sets. It, like *LAR*, uses the available position information of the destination from a route established earlier to implement a restricted directional flooding among grids. Consequently it is robust against position inaccuracy since it uses grid-by-grid routing and expected region concept. Although *GRID* has strong route maintenance capability and is very robust as regards node mobility, it is like any other protocol that depends on route setup before sending the data packets in the sense that the failure of a single node might result in packet loss and the setting up of a new route. Moreover, the authors in [21] did not elaborate on the route maintenance required when a grid remains empty after its leader and only node leaves it [16]. Thus, its robustness is considered to be medium.

GRID's implementation complexity is considered to be medium due to its dealing with the area as grids. Its scalability is high due to its use of restricted directional flooding and the delegating of the search responsibility to gateway hosts. Its packet overhead is considered to be low due to the reduced number of small routing packets. *GRID* uses a specific field to detect duplicate request packets from the same source, so endless flooding of the same request can be avoided; i.e., it is loop free. On the other hand, *GRID* uses long paths since packets are forced to be routed through grids' leaders.

Finally, it is better to implement *GRID* in dense networks because of its routing in a grid-by-grid manner through grids' leaders. So the number of packets related to route search is insensitive to the network density. On the contrary, the cost decreases slightly as the host density increases, since routes become more stable with denser hosts. On the other hand, if it is implemented in sparse networks, each node will be the gateway of its grid, and *GRID* may become like native *LAR*; consequently, it will consume network resources in dividing the area into grids and electing gateways without any benefit.

3.3.2 *TERMINODES*

TERMINODES [9] is an example of hierarchical routing protocols. *TERMINODES* presents a two-level hierarchy within which, if the destination is close to the sender (in terms of number of hops), packets will be routed based on a proactive distance vector. Greedy routing is used in long-distance routing [17].

TERMINODES addresses the following objectives: scalability (both in terms of the number of nodes and geographical coverage), robustness, collaboration and simplicity of nodes [16].

This routing scheme is a combination of two protocols called *Terminode Local Routing (TLR)* and *Terminode Remote Routing (TRR)*. *TLR* is a mechanism that allows for the reaching of destinations in the vicinity of a terminode and does not use location information for making packet forwarding decisions. *TRR* is used to send data to remote destinations and uses geographic information; it is the key element for achieving scalability and reduced dependence on intermediate systems. The major novelty is the *Anchored Geodesic Packet Forwarding (AGPF)* component of *TRR*. This is a source-path-based method designed to be robust for mobile networks: instead of using traditional source paths, that is lists of nodes, it uses anchored paths. An anchored path is a list of fixed geographical points, called anchors. The packet loosely follows the anchored path. At any point, the packet is sent in the direction of the next anchor in the anchored path by applying geodesic packet forwarding. When a terminode finds that the next anchor geographically falls within its transmission range, it deletes this from the anchored path and sends the packet in the direction of the new next anchor. This is repeated until the packet is sent in direction of the final destination [16].

The authors of [9] showed by means of simulations for mobile Ad-Hoc networks composed of several hundreds of terminodes 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. They also demonstrated the benefits of the combination of *TLR* and *TRR* over an existing protocol that uses geographical information for packet forwarding [13]. However, with the use of greedy routing in long distance routing, *TERMINODES* inherits the problems associated with it.

Discussion

TERMINODES provides a hierarchical approach to position-based Ad-Hoc routing. For long distance-routing it uses a greedy approach and therefore has characteristics similar to those of greedy forwarding. However, due to the usage of a non-position-based approach at the local level, it is more tolerant of position inaccuracy. As with other greedy forwarding protocols, in *TERMINODES* all nodes maintain a one-hop neighbor table; it uses all-for-some location service. Hence, a given node will be inaccessible upon the failure of a subset of the nodes; its location service has a medium robustness. Moreover it may fail to find the optimum route and has higher delivery rates for dense graphs.

TERMINODES's 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 using the two-level hierarchy approach, *TERMINODES* is considered to have medium implementation complexity. Such an approach is scalable since it does not require routing discovery and maintenance in long-distance routing. Moreover, it has low packet overhead due to its small number of small-size packets. *TERMINODES* is considered to be a loop-free algorithm [16] since it always forces the message to make a step closer to the destination.

3.3.3 LABAR

Location Area Based Ad-Hoc Routing for GPS-Scarce Wide-Area Ad-Hoc Networks (LABAR) [6] is a hybrid virtual backbone and geographical location area based Ad-Hoc routing. The authors outlined that using *GPS* can increase the cost and power consumption of small mobile nodes. Thus, *LABAR* requires only a subset of nodes (called G-nodes) to know their exact location, forming location areas around them. G-nodes are interconnected into a virtual backbone structure to enable the efficient exchange of information for the mapping of the *IP* addresses to locations. Nodes that are not enabled with *GPS* equipment are called S-nodes.

Routing in *LABAR* consists mainly of three steps: zone formation, virtual backbone formation and directional routing. The first step of *LABAR* deals with forming the zones; i.e., making the decision as to which S-nodes should belong to which G-nodes. It is assumed that all G-nodes start the zone formation algorithm at the same time to acquire S-nodes. If an S-node has already been attached to a G-node, then the request message is ignored by the S-node. Upon being included in a zone, an S-node initiates the zone formation algorithm on its own in order to draw more S-nodes and form its neighborhood into its zone. By the end of this step, all S-nodes will belong to a G-node and G-nodes will know the IDs of their zone's S-nodes. The second step is to create an easy-to-manage virtual backbone for relaying the position information of nodes. G-nodes in the virtual backbone are responsible for resolving the *IP* addresses into geographical locations. To connect zones and get the virtual backbone to function, a G-node called the root sends connect messages to its adjacent zones. If the particular adjacent zone is not connected yet to the backbone, then it will be added to the backbone. Figure 9 shows an example of such a virtual backbone.

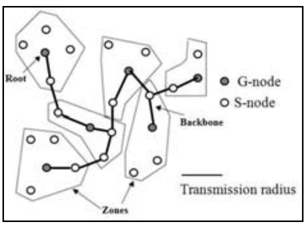

Figure 9
Example of virtual backbone in *LABAR*

The last step is directional routing. The source node queries the source G-node node to map the destination *IP* address into the geographical location area of the destination. Then the source G-node determines the vector pointing from its own location to the destination's location. The resulting vector's direction is compared to each of the adjacent zones' directions and distances to determine the neighboring zone that will be used in relaying the data to the destination. Now, the source G-node will instruct the source node on how to route the packet inside the zone to reach the next zone with the least number of hops. The node that receives the packet in the neighboring zone will route the packet to the next zone by consulting its zone's G-node (which will consume time). In the case of a failure in the directional route (determined for example through expired hop counters), the source zone will be informed about the failure and the virtual backbone will be used to relay the packets.

LABAR is a combination of proactive and reactive protocols since the virtual backbone structure is used to update location information between G-nodes (in a proactive manner), while user packets are relayed using directional routing towards the direction zone of the destination. One of the important advantages of *LABAR* is the reduction of cost and power consumption through the relaxation of the *GPS*-equipment requirement in each node.

Discussion

LABAR is a hierarchal protocol since it uses zone-based routing. In *LABAR* the virtual backbone structure is used to update location information between G-nodes in a proactive manner; the used location service type is some-for-all. Generally, the robustness of such approaches is medium, since the position of a node will become unavailable if a subset of the nodes failed. *LABAR* exhibits some properties of greedy forwarding such as high scalability, low packet overhead and its suitability to be implemented in dense networks. *LABAR* is tolerable to position inaccuracy through its relaying the user packets towards the direction of the destination's zone, not towards its exact position.

In the case of a failure in the directional route of *LABAR*, the virtual backbone will be used to relay the packets; i.e., *LABAR*'s robustness is high since a failure of a single intermediate node does not prevent the packet from reaching its destination. *LABAR*'s implementation complexity is considered to be medium because of its use of zones. *LABAR* was not considered a loop-free protocol since it uses directional flooding and does not use any technique to indicate that a specific packet has been received earlier by a specific node. However, the expired hop counters can be used to stop the loops after a while.

One may think that *LABAR*'s probability of finding the optimal path is medium since it uses greedy routing; however, simulations have shown that *LABAR* usually uses a long path which sometimes exceeds double the length of the optimal path. This may be owing to using directional route towards the direction of destination's zone, not toward the exact position of it. So, *LABAR*'s probability of finding the optimal path is considered to be low.

4 Summary of the Selected Protocols

Table 1 summarizes the discussed protocols together with the evaluation criteria used.

Table 1

Characteristics of the presented forwarding strategies

Metric	**LS type**	**LS robustness**	**FS type**	**FS toleration to position inaccuracy**	**FS robustness**	**FS implement. complexity**	**FS scalability**	**FS packet overhead**	**Loop free**	**Optimal path**	**Density**
MFR	All-for-Some	Medium	Greedy (progress)	TR	Medium	Low	High	Low	Yes[8]	Medium	Dense
DIR	All-for-Some	Medium	Greedy (direction)	TR	Medium	Low	High	Low	No[8]	Medium	Dense
GPSR	All-for-Some	Medium	Greedy+ perimeter	TR	Medium	Medium	High	Low	Yes[16]	Medium	Both
ARP	All-for-Some	Medium	Greedy (distance + angle)	TR	Medium	Low	High	Low	No (only local ones)	Medium	Both
I-PBBLR	-	-	Greedy (progress + direction)	No beacons	Medium	Low	High	Low	Yes[23]	Medium	Dense
POSANT	-	-	Multiple Greedy (pheromone)	Zones and pheromone	Medium	Medium	Medium	Medium	Yes[18]	High	Both
DREAM	All-for-All	High	RDF	ER	High	Low	Medium	Medium	No[8]	High	Both
LAR	-	-	RDF	ER	Low	Low	Medium	Medium	No[8]	High	Both
LARWB	-	-	RDF	ER	Medium	Low	Medium	Medium	No	High	Both
MLAR	-	-	RDF	ER	Medium	Low	Medium	Medium	Yes[19]	Low	Both
GRID	-	-	Hierarchical	Grid-by-grid routing	Medium	Medium	High	Low	Yes	Low	Dense
TERMIN-ODES	All-for-Some	Medium	Hierarchical	Short-distance routing range	Medium	Medium	High	Low	Yes[16]	Medium	Dense
LABAR	Some-for-All	Medium	Hierarchical	Zones	High	Medium	High	Low	No	Low	Dense

Abbreviations: LS: Location Service. FS: Forwarding Strategy. RDF: Restricted Directional Flooding.
TR: Transmission Range. ER: Expected Region.

5 Directions of Future Research

In this paper we have shown that there are many approaches to performing position-based packet forwarding. However, there still exist a number of issues and problems that need to be addressed in future research.

Position-based protocols make it possible to have larger networks without scalability problems. However, geographical routing also offers attackers new opportunities, especially due to the fact that most protocols broadcast position information in the clear, allowing anyone within range to receive it. Hence, node position can be altered, making other nodes believe that it is in a different position. This may make nodes believe that the attacker is the closest node to the destination and choose it as the next hop. Consequently, this attacker will be able to alter or drop packets.

Thus, it is worthwhile that more intensive work be done to secure position-based routing protocols to be able to defend against several attacks, not only from malicious nodes, but also from the compromised ones. Additionally, location privacy is one of the most major issues which need to be addressed, especially the fact that location privacy is hard to achieve when a node identifier can be immediately associated with its position.

Geographical routing protocols depend strongly on the existence of distributed scalable location services which are able to provide the location of any host at any time throughout the entire network. Hence, research should consider the scalability point upon developing new location services. Also, the most common way to enable nodes to know their locations is by equipping them with *GPS*. To decrease the cost and power consumption of small mobile nodes, other techniques for finding relative coordinates should be discussed.

We also need more concentration on power-conscious routing for saving network energy through the development of protocols that have as many sleeping nodes as possible and designing sleep period schedules for each node. Also, more studies should concentrate on *Quality of Service (QoS)*, geocast and multicast position-based routing.

Most routing protocols (not only position-based) consider nodes as neighbors if the Euclidean distance between them at most equals the transmission radius, which is the same for all nodes in the network. However, irregular transmission radius of a node (due to obstacles or noise), unidirectional links and different nodes' transmission radii should be taken into consideration. Moreover, many applications have nodes distributed in 3-Dimensional space, and little research has yet been done in this field.

Another issue that needs to be addressed is enabling connectivity among the individual Ad-Hoc networks, as well as the connectivity of any Ad-Hoc network to the Internet. This will, most likely, require the usage of hierarchal approaches to achieve scalability. This field has already been begun, but it needs further investigation.

Summary and Conclusions

Efficient routing among a set of mobile hosts is one of the most important functions in Ad-Hoc wireless networks. Many points should be taken into consideration when developing a routing protocol; some of these points are high delivery rate, reduced number of hops, small flooding ratio, small end-to-end delay and low power consumption. This survey has presented the current state of unicast position-based Ad-Hoc routing and provided a qualitative evaluation of the presented approaches. At the end, we identified a number of research opportunities which could lead to further improvements in position-based Ad-Hoc routing.

Forwarding techniques based on position information were classified into three distinct categories. Greedy routing does not require the maintenance of explicit routes; instead, it works by forwarding a single copy of data packet towards the destination. If a local maximum is encountered, a repair strategy can be used to avoid dropping the packet. After the comparison of the existing solutions we can conclude that the greedy packet forwarding is an efficient approach that scales well even with highly dynamic networks, and it is a promising strategy for general purpose position-based routing. However, it is not guaranteed to find the optimal path, and it may not find a path at all.

In restricted directional flooding the packets are broadcast 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. Restricted directional flooding has higher packet overhead and less scalability; however, its opportunity of finding the shortest path is higher. Using restricted directional flooding to set up a route in an efficient manner (such as in *LAR*) increases the probability of finding the optimal path and is suitable for cases that require a high volume of data transmissions. However, when it is used to forward the data packets themselves (such as in *DREAM*) it will be more suitable for situations where a small number of packets need to be transmitted very reliably.

Using hierarchical approaches increases the approach scalability. This may be done through the usage of zone-based routing, dominating sets, or by means of a position-independent protocol at the local level and a greedy variant at the long-distance level.

Security has recently gained a lot of attentions in topology-based routing protocols and many attempts to propose end-to-end security schemes have been made. However, it is obvious from the analysis that few research efforts have addressed position-based security issues. Finally, a few researchers have considered the power efficiency metric while developing their protocols.

Acknowledgement

This work was done under the VotF University Malaya fund no. FS132/2008C, PPP University Malaya fund no. PS091/2009A and PPP University Malaya fund no. PS410/2010B.

References

[1] B. Karp, H. Kung: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, in Proceedings of 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 2000), Boston, Massachusetts, USA, 2000, pp. 243-254

[2] C. Perkins, E. Royer: Ad Hoc On-demand Distance Vector Routing, in Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, IEEE, 1999, pp. 90-100

[3] D. Johnson, D. Maltz: Dynamic Source Routing in Ad Hoc Wireless Networks, Mobile Computing, Vol. 353, Kluwer Academic Publishers, 1996, pp. 153-181

[4] E. Kranakis, H. Singh, J. Urrutia: Compass Routing on Geometric Networks, in Proceedings of 11th Canadian Conference on Computational Geometry, Vancouver, 1999, pp. 51-54

[5] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan: Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks, in Proceedings of 7th Annual International Conference on Mobile Computing and Networking (MOBICOM 2001), Rome, Italy, ACM, 2001, pp. 272-287

[6] G. Zaruba, V. Chaluvadi, A. Suleman: LABAR: Location Area Based Ad Hoc Routing for GPS-Scarce Wide-Area Ad Hoc Networks, in Proceedings of 1st IEEE International Conference on Pervasive Computing and Communications (PerCom'03), 2003, pp. 509-513

[7] H. Takagi, L. Kleinrock: Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals, IEEE Transactions on Communications, Vol. 32, No. 3, 1984, pp. 246-257

[8] I. Stojmenovic, X. Lin: Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks, IEEE Transactions on Parallel and Distributed Systems, Vol. 12, No. 10, 2001, pp. 1023-1032

[9] L. Blazevic, L. Buttyan, S. Capkum, S. Giordano, J. Hubaux, J. Le Boudec: Self-Organization in Mobile Ad-Hoc Networks: the Approach of Terminodes, IEEE Communication Magazine, Vol. 39, No. 6, 2001, pp. 166-174

[10] L. Feeney, M. Nilsson: Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment, in Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), Vol. 3, USA, 2001, pp. 1548-1557

[11] L. Qabajeh, M. L. Mat Kiah, M. Qabajeh: A Qualitative Comparison of Position-based Routing Protocols for Ad-Hoc Networks, International Journal of Computer Science and Network, Vol. 9, No. 2, 2009, pp.131-140

[12] M. Marina, S. Das: On-Demand Multipath Distance Vector Routing in Ad Hoc Networks, in Proceedings of 9th International Conference on Network Protocols (ICNP 01), 2001, pp. 14-23

[13] M. Mauve, J. Widmer, H. Hartenstein: A Survey on Position-based Routing in Mobile Ad-Hoc Networks, IEEE Network, Vol. 15, No. 6, 2001, pp. 30-39

[14] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia: Routing with Guaranteed Delivery in Ad Hoc Wireless Networks, in Proceedings of 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 1999, pp. 48-55

[15] S. Basagni, I. Chlamtac, V. Syrotiuk, B. Woodward: A Distance Routing Effect Algorithm for Mobility (DREAM), in Proceedings of 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Dallas, TX, USA, 1998, pp. 76-84

[16] S. Giordano, I. Stojmenovic, L. Blazevic: Position-based Routing Algorithms for Ad Hoc Networks: A Taxonomy. In Cheng, X., Huang, X., Du, D.Z: Ad hoc wireless Networking, Kluwer, 2003, pp. 103-136 http://www.site.uottawa.ca/~ivan/routing-survey.pdf

[17] S. Kalhor, M. Anisi, A. Haghighat: A New Position-based Routing Protocol for Reducing the Number of Exchanged Route Request Messages in Mobile Ad-hoc Networks, in Proceedings of 2nd International Conference on Systems and Networks Communications (ICSNC 2007), IEEE, 2007, p. 13

[18] S. Kamali, J. Opatrny: POSANT: A Position-based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks, in Proceedings of 3rd International Conference on Wireless and Mobile Communications (ICWMC'07), IEEE, 2007

[19] S. Nanda, R. Gray: Multipath Location-aided Routing in 2D and 3D, IEEE Wireless Communications and Networking Conference (WCNC 2006), Vol. 1, 2006, pp. 311-317

[20] V. Giruka, M. Singhal: Angular Routing Protocol for Mobile Ad-hoc Networks, in Proceedings of 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW’05), 2005, pp. 551-557

[21] W. Liao, Y. Tseng, J. Sheu: GRID: A Fully Location-Aware Routing Protocols for Mobile Ad Hoc Networks, Telecommunication Systems, Vol. 18, 2001, pp. 61-84

[22] X. Wu: VPDS: Virtual Home Region-based Distributed Position Service in Mobile Ad Hoc Networks, in Proceedings of 25th IEEE International Conference on Distributed Computing Systems (ICSCS 2005), 2005, pp. 113-122

[23] Y. Cao, S. Xie: A Position-based Beaconless Routing Algorithm for Mobile Ad Hoc Networks, in Proceedings of International Conference on Communications, Circuits and Systems, Vol. 1, IEEE, 2005, pp. 303-307

[24] Y. Ko, N. Vaidya: Location-aided Routing (LAR) in Mobile Ad Hoc Networks, Wireless Network (WINET), Vol. 6, No. 4, ACM, 2000, pp. 307-321