Wireless Netw
DOI 10.1007/s11276-006-9955-y

Location-aware routing protocol with dynamic adaptation of request zone for mobile ad hoc networks

Tzay-Farn Shih · Hsu-Chun Yen

Published online: xxx
© Springer Science + Business Media, LLC 2006

UNCORRECTED PROOF

Abstract One possibility direction to assist routing in Mobile Ad Hoc Network (MANET) is to use geographical location information provided by positioning devices such as global positioning systems (GPS). Instead of searching the route in the entire network blindly, position-based routing protocol uses the location information of mobile nodes to confine the route searching space into a smaller estimated range. The smaller route searching space to be searched, the less routing overhead and broadcast storm problem will occur.

In this paper, we proposed a location-based routing protocol called LARDAR. There are three important characteristics be used in our protocol to improve the performance. Firstly, we use the location information of destination node to predict a smaller triangle or rectangle request zone that covers the position of destination in the past. The smaller route discovery space reduces the traffic of route request and the probability of collision. Secondly, in order to adapt the precision of the estimated request zone, and reduce the searching range, we applied a *dynamic adaptation of request zone technique* to trigger intermediate nodes using the location information of destination node to redefine a more precise request zone. Finally, an *increasing-exclusive search approach* is used to redo route discovery by a progressive increasing search angle basis when route discovery failed. This progressive increased request zone and exclusive search method is helpful to reduce routing overhead. It guarantees that the areas of route rediscovery will never exceed twice the entire network. Simulation results show that LARDAR has lower routing cost and collision than other protocols.

Keywords MANET · Location-aware · Position-based routing protocol · GPS

1 Introduction

The advances in wireless communication and portable computing devices have made mobile computing possible. There are currently two variations of mobile wireless networks. The first is known as the infrastructure networks, which have fixed base stations. The mobile nodes connect to, and communicate with, the nearest base station within their communication range. A handoff occurs from the old base station to the new when the mobile unit travels out of range of one base station and into the range of another. The second type of mobile wireless network is the infrastructureless mobile network, commonly known as mobile ad hoc network (MANET). A mobile ad hoc network is a collection of mobile nodes that are dynamically and arbitrarily located so that the interconnections between nodes are capable of changing on a continual basis. Unlike conventional wireless networks, ad hoc networks are wireless networks with no fixed routers, hosts, or wireless base stations. In an ad hoc network, there is no dedicated base station to manage the channel resources for each network node. Carefully designed distributed medium access techniques must be used for channel resources. Nodes of these networks function as routers, which discover and maintain routes to other nodes in the network.

Tzay-Farn Shih was with Department of Electrical Engineering, National Taiwan University.

T.-F. Shih
Department of Computer Science and Information Engineering, Chaoyan, University of Technology, Taichung, Taiwan 41349, R.O.C.

H.-C. Yen (✉)
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 106, R.O.C.
e-mail: yen@cc.ee.ntu.edu.tw

In recent years, routing protocol in ad hoc networks as a research topic has started to receive more attention. Many graph-based routing protocols have been proposed for mobile ad hoc networks [1–8]. These protocols use the known connectivity relation with its neighbors to do route discovery blindly. Blindly searching a whole network produces huge routing traffic and collision. It not only wastes a large portion of wireless bandwidth, but also induces higher route construction time. In order to reduce the routing overhead, a novel routing protocol, called position-based routing protocol, which using the location information of mobile nodes to assist the routing task has attracted more attention. The availability of small, low-power global positioning system receivers for calculating relative coordinates make it possible to apply position-based routing algorithms in mobile ad hoc networks. Using location information to improve performance of a mobile computing system routing has been suggested in [16–26].

The position-based routing algorithms require that information about the physical position of the participating nodes be available. Usually, each node determines its own geographical location through the use of GPS or some other type of positioning device [9–15]. A survey of these methods can be found in [13]. 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 nodes. Position-based routing thus does not require the maintenance of routes by transmitting 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 [17, 22].

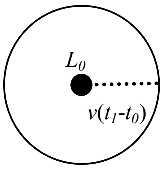
This paper investigated the routing problem in MANET by using the location information of mobile nodes. We proposed an approach named ***L**ocation-**A**ware **R**outing protocol with **D**ynamic **A**daptation of **R**equest zone* (LARDAR). This uses the geographical location to define a route searching space, called a request zone, for the specified destination node. Using the destination node's location, location information obtained time (i.e., the timestamp when the destination's location was collected), and velocity, we can predict a smaller triangle or rectangle request zone which covers the possible location of destination in the past time. The smaller route discovery space reduces the traffic of route request and the probability of collision. In order to refine the estimated request zone and reduce the searching range, we also introduce a *dynamic adaptation of request zone technique* in our protocol to trigger intermediate nodes using its more fresh location information of destination node or the information provide by precedent node to redefine a more precise request zone to improve the performance of routing protocol. In our protocol, an *increasing-exclusive search approach* is used to redo route discovery by a progressive increasing searching angle basis when previous route request failed. This progressive increase request zone and exclusive search method is helpful to reduce routing overhead. Simulation results show that our LARDAR has lower route setup time, routing cost and collision than LAR proposed in [20].

The rest of the paper is organized as follows. Section 2 presents some proposed protocols. Section 3 shows the motivation of our LARDAR. Our protocol is described in Section 4. Section 5 presents the simulation results. Finally, we give a conclusion in Section 6.

2 Related work

Y. Ko and N. H. Vaidya proposed a location-aided routing (LAR) in mobile ad hoc Networks [20]. LAR tries to exploit location information in the route discovery process. In the discovery procedure of LAR, source node S uses the location information of destination node D to estimate the region that the destination node expects to appear, they called it the expected zone. Given t_1, node S knows that node D was at location $L_0(x_0, y_0)$ at time t_0, and D's average moving speed is v. Then, S may determine the expected zone to be a circular region with a radius $v(t_1 - t_0)$ and centered at location $L_0(x_0, y_0)$, see Fig. 1. If S does not know the location of D, the expected zone is set equal to the entire network. Using the expected zone and the location of S, source S can define a request zone as show in Fig. 2(a) and (b) when S is outside or within the expected zone respectively. To increase the probability that the route request will reach D, the request zone should include the expected zone. LAR is based on flooding with one modification as follows. In LAR, the route request packet is only forwarded by nodes that are within the request zone. For instance, in Fig. 2(a), if node I_1 receives a route request from S, it relays the request to its neighbors since it is within the request zone. When I_2 receives a request packet, it discards this packet immediately, as it is outside the request zone. This is better than a blind search in the whole network as traditional flooding algorithms do and can save on routing costs. The route discovery success ratio depends on the request zone. If S does not know D's location, the LAR algorithm reduces to the basic flooding algorithm by blindly searching the whole network.

Fig. 1 The expected zone



UNCORRECTED PROOF

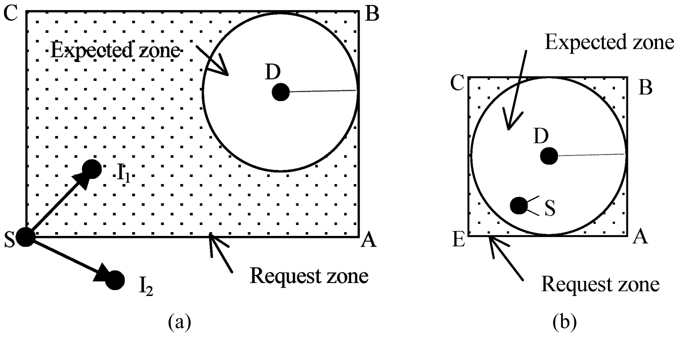


Fig. 2 (a) S is outside the expected zone. (b) S is within the expected zone

Another request region confined algorithm, A Distance Routing Effect Algorithm for Mobility (DREAM), is proposed by S. Basagni et al. [19]. In DREAM, each node maintains a location database that stores location information about nodes that are part of the network. Each node regularly floods packets to update the location information maintained by the other nodes. The accuracy of such an entry depends on its age. A node can control the accuracy of its location information available to other nodes by location update frequency and location update travel distance. Concerning the maintenance of the location database, DREAM can be classified as proactive routing protocol. The route construction is based on an on demand fashion, like a reactive routing algorithm. When the source node needs to transmit a message, it uses the location information of the destination node to estimate the direction of the destination node, and then forwards the message to all its one-hop neighbors within this confined direction. Each neighbor node in the confined direction repeats the same procedure until the message reaches the destination node if it is reachable. Assume that source node S wants to transmit a message to destination node D at time t_1. If, at time t_1, S knows D's location obtained at time t_0, and D's average speed v in the interval from t_0 to t_1, then S can use the location information of D to estimate the direction of D, as depicted in Fig. 3. S transmits the message to its one-hop neighbors I_1 and I_2, because they are confined in the direction $[\theta-\alpha, \theta+\alpha]$. Neighbor node I_1 and I_2 repeats the same procedure until node D is reached. If r is the maximum distance that D can travel from time t_0 to t_1, and D is the distance between S and D, the value of α can be computed as follows:

$$\alpha = \sin^{-1}\left(\frac{r}{d}\right), \quad \text{where } r = v(t_1 - t_0) \qquad (1)$$

If $r > d$, then D can be in any direction. In this case, the value of α will be set to π. If node S does not know the location of node D, then S transmits the message by a *Recovery* procedure.

Y. Yu et al. proposed a Geographical and Energy Aware Routing protocol (GEAR) to reduce route search overhead [29]. GEAR uses energy aware and geographical informed neighbor selection to route a packet towards the target region. A recursive geographic forwarding or restricted flooding algorithm is used to disseminate the packet inside the destination region. This protocol attempts to balance energy consumption and thereby increase network lifetime.

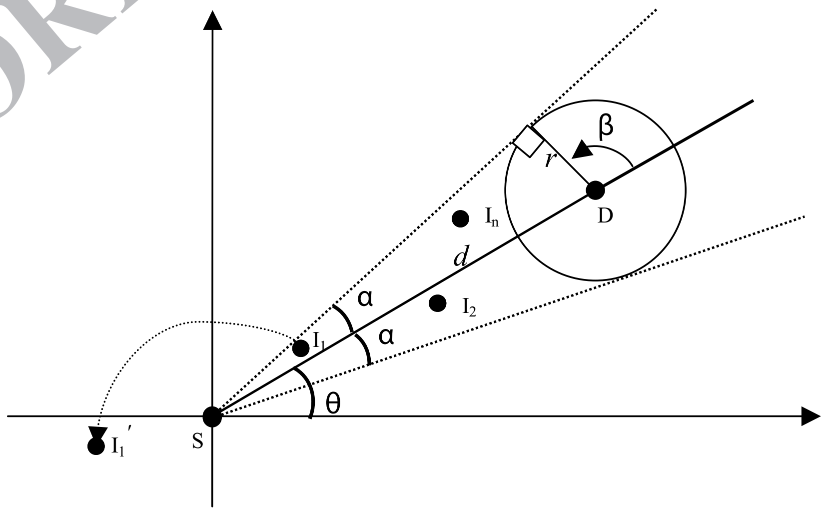


Fig. 3 The expected direction of destination node D at time t_1

UNCORRECTED PROOF

3 Motivation

We propose a protocol to improve the routing performance according to the drawbacks of LAR, DREAM and GEAR in the following aspects.

In LAR algorithm, the request zone is defined to be the smallest rectangle that includes the source node and the expected zone, such that the sides of the rectangle are parallel to the X and Y axes. In order to reduce the route searching space, we defined a triangle-shaped request zone. The triangle-shaped request zone is smaller than the request zone defined in LAR and guarantees to cover the source node and the expected zone. When the route discovery fail in the first attempt, LAR redo route discovery by flooding that enlarges the request zone to entire network rapidly. We adopted a *progressive search angle increasing mechanism* to avoid huge routing traffic and collision, caused by the flooding policy used by LAR. On the other hand, we introduced an exclusive searching method to limit the number of participant nodes in the route rediscovery under an upper bound of $2n$, where n is the total number of nodes in the network. The exclusive searching method guarantees that the number of route request packets be forwarded in the reroute procedure never exceeds twice the number of request packets relayed by flooding.

DREAM 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). Due to the communication complexity of position updates, DREAM is inappropriate for large scale and general purpose ad hoc networks. The routing success rate of DREAM is not only closely relying on fresh location information of the recipient but also the precise location database in each intermediate node that participates in routing. The performance impacted by more location error facts than LAR. Considering Fig. 3 as an example, when S wants to send message to D at time t_1, it estimates the direction of D and selects one hop neighbors that locate in this direction by the location database in it. If the location information of I_1 in node S is out-of-date, assume that it is moved to I'_1 before time $t_{1,}$ then S will make a mistake to choose I_1 as a forwarding node and forwards the message to it. As Fig. 3 shows, the message is sent away from the destination node in the opposite direction. If the same situation occurred in the other node, it causes a vicious circle and the algorithm will fail. The result reveals that DREAM can not guarantee to enclose the forwarding node in the direction to destination node. If an intermediate node has no idea about the location of D, the route discovery will fail and the message is discarded in this node. The deep dependence on location information of all routing nodes increases the probability of routing failure. To overcome this problem, our protocol LARDAR applies a *dynamic adaptation of request zone approach*. This approach uses the newest location information of destination node carried in the route request packet or stored in the intermediate node's location database to adapt request zone dynamically. We will detail our protocol in next section.

In GEAR, the route request packet is forwarded according to an energy-aware neighbor computation that selects a next-hop node based on the learned cost or estimated cost [29]. For the definition of learned cost or estimated cost in GEAR, the route discovery is highly depending on the information of neighbor nodes such as location, energy level, and learned cost, etc.. GEAR is assumed to design for static (i.e., immobile) sensor networks. In the immobile topology of sensor networks, the neighbors and neighbors' location are fixed. The update of the information of neighbors is infrequently and easily. The overhead of control packet might be negligible in the static topology. On the contrary, applying GEAR in high mobility MANET, a huge control overhead will be induced for the neighbors' information update that might degrade the performance and scalability. It is not trivial for the neighbors' information update in a mobile ad hoc network. In the high mobility MANET, the route discovery of GEAR is more sensitive to the error of neighbors' information, e.g., location error, learned cost or estimated cost error. The incorrect or loss of neighbor information will break down the route discovery. Another problem of GEAR is its energy balance technique might increase the average path length. In this case, there are more nodes burning their energy and the transmission latency increases. The incorrect or loss of neighbor information does not crash our LARDAR because the route request can be do by flooding in current hop and the confined search will work again while the neighbor information is acquired. The triangle request zone of LARDAR forces the request packet to propagate as straight to destination as possible. It has a higher chance to find a shorter path.

4 The LARDAR routing protocol

4.1 The definition of expected zone

In this section, we discuss the definition of expected zone based on the same assumption described in Section 2. There are two methods to define the expected zone. The first method is the same as LAR that is described in Section 2. Considering the first definition, source node S estimates an expected zone that is a circular region of radius $v(t_1 - t_0)$, centered at location $L_0(x_0, y_0)$. The circle is a region that potentially contains destination node D between time t_0 and t_1. When the route request arrived the original location of D, some time passed, say Δt. In this time interval, node D might travel outside the expected zone. In order to calculate a more exact expected zone, we must take the time interval Δt into account. Another

UNCORRECTED PROOF

definition of expected zone can be defined as a circular region with a radius $v[(t_1 + \Delta t) - t_0]$. This method is more precise than the first. In this method, a challenge is how to determine the value of Δt. For simplicity, Δt can be set to half of the round trip time between S and D. Another more precise and complex method to get the value of Δt is as follows: When S received a packet from D, S adds the location information of D and the transmission time of this packet from D to S into its routing table. If S needs to calculate an expected zone for D, we can let Δt as the transmission time from D to S that is recorded in the routing table of S.

4.2 The definition of request zone

According to the relative location between the source node and the expected zone, the definition of request zone can be classified as follows:

- **Source node is outside of expected zone—TRIANGLEZONE:** If source node S is outside of the expected zone, we define the request zone to be the smallest isosceles triangle, named TRIANGLEZONE, which includes the current location of S and the expected zone. There are two advantages to define a TRIANGLEZONE request zone. Firstly, a TRIANGLEZONE restrains the route request packet to forward in a narrower space. It means that the request is forced to propagate as straight to destination as possible. It is good for providing a higher chance to select a shorter route. Secondly, the area of TRIANGLEZONE is less than the area of rectangle request zone defined by LAR. It means that our algorithm confined the route search to a smaller space. For instance, in Fig. 4(a), the area of TRIANGLEZONE whose corners are S, E and G is less than the area of rectangle request zone with the corners S, A, B and C that is defined by LAR. The area of TRIANGLEZONE ΔSEG can be computed as follows:

$$A_{\Delta SEG} = (d + r)^2 \tan\alpha = \left(\sqrt{(x_d - x_s)^2 + (y_d - y_s)^2} + r\right)^2 \tan\alpha \qquad (2)$$

The area of rectangle $\square$SABC can be represented by the following equation:

$$\mathrm{A}_{\square \mathrm{SABC}} = (x_d - x_s + r)(y_d - y_s + r) \qquad (3)$$

Using Eqs. (2) and (3), we define the request zone reduction ratio between ΔSEG and $\square$SABC as below:

$$R = 1 - (\mathrm{A}_{\Delta \mathrm{SEG}}/\mathrm{A}_{\square \mathrm{SABC}}) = 1 - \frac{\left(\sqrt{(x_d - x_s)^2 + (y_d - y_s)^2} + r\right)^2 \tan\alpha}{(x_d - x_s + r)(y_d - y_s + r)} \qquad (4)$$

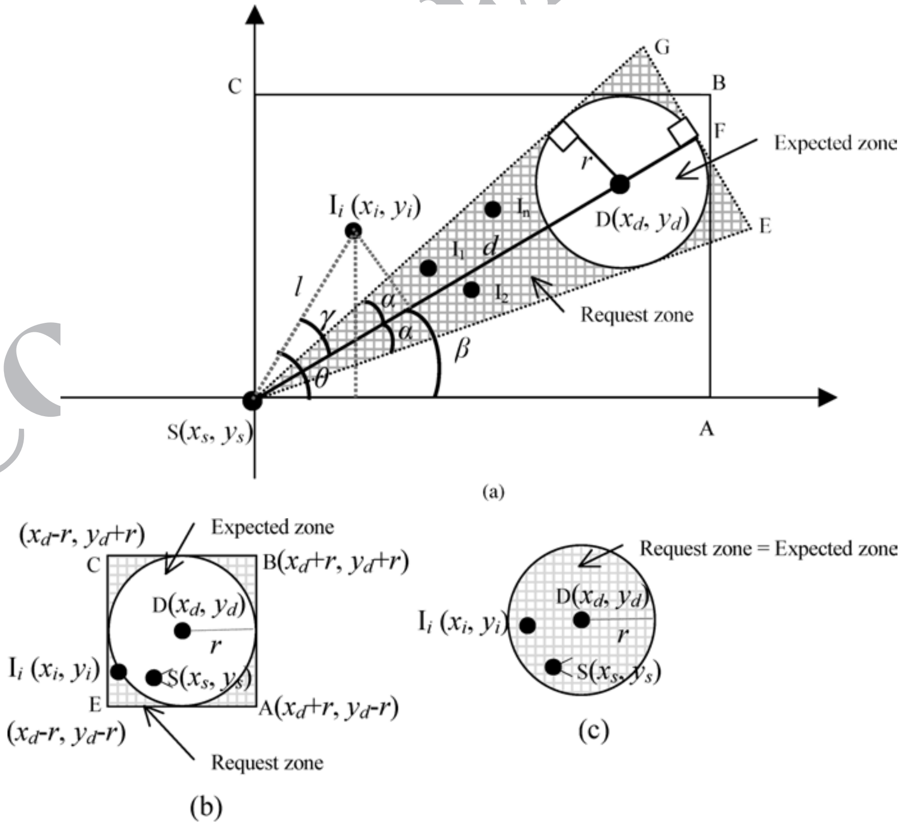


Fig. 4 (a) TRIANGLEZONE, source node S outside the expected zone. (b) RECTZONE, S within the expected zone. (c) CIRCLEZONE, S within the expected zone

UNCORRECTED PROOF

UNCORRECTED PROOF

By Eq. (4), we find that the request zone reduction ratio R is inverse proportion to the searching angle α. The smaller the search angle, the smaller the search space is probing, and a lower routing overhead is required. Therefore, the routing cost and the routing success rate can be controlled by adapting the search angle.

In the route request procedure, the location of S, the distance between S and F, angle α and β must be included into the route request message, we define request zone to be a TRIANGLEZONE (see Fig. 4(a)). When a node receives the request, it uses the information for determining if it is a forwarding node. The detailed operation is described in Section 4.3.

If there are holes in the triangle zone, the route discoveries are likely to be done multiple times, which in turn increases the routing overhead and the route setup time. To overcome the problem caused by hole, a *hole detection technique* is included in the TRIANGLEZONE computation. The detail of this technique is as follows. In the TRIANGLEZONE computation process at each node, the node checks if there are neighbors locate within the TRIANGLEZONE by using the neighbors' location information that are recorded in its neighbor table. If there is no neighbor node within the TRIANGLEZONE, it means that a hole exists and the route request packet can not be forwarded to the destination in this request zone successfully. It will increase the search angle and recalculate a TRIANGLEZONE until there is no hole exists or the search angle is greater than the threshold. The route request packet is send only when there is no hole exists. Another method to reduce the probability of hole is applying directional antenna technique. If all nodes equip with directional antenna, their transmission radius will increase. It means that the root node of the TRIANGLEZONE (i.e., sender) will has higher chance to cover neighbor within its transmission range. The routing overhead and the routing success rate can be controlled by correctly select the value of angle increment and threshold. They are heuristic and could be improved by genetic algorithm. Hence, the problem of hole can be reduced by the above techniques.

- **Source node is within the expected zone—**

If source node S is within the expected zone, there are two ways to define the request zone.

RECTZONE: The first one is the same as LAR that defines the request zone to be the smallest rectangle, called RECTZONE, which includes S and the expected zone. The example in Fig. 4(b) is a RECTZONE with corners at point A, B, C and E. In this case, the coordinates of B and E must be included into the route request packet. The coordinates of B and E are used for the forwarding node membership determination by next hop node in the route request process.

CIRCLEZONE: The second method defines the request zone to be a circle, called CIRCLEZONE, which is equal to the expected zone, shown in Fig. 4(c). It is quite obvious that the CIRCLEZONE confines the routing space to a smaller area than RECTZONE. In the route request procedure, the necessary information to be carried in the request message for forwarding node membership judgment is the location of destination and the radius of the CIRCLEZONE.

While a sender that used a TRIANGLEZONE in last routing process fails to find the route to a destination within a timeout interval, it enlarges the TRIANGLEZONE by increasing the search angle with an angle increment. If a route is not discovered when the search angle exceeds a threshold, the source node expands the request zone to the entire network and reroutes again. If source node has no location information of destination node, it defines the request zone to the entire network and initiates a route discovery by flooding.

4.3 Determining the membership of forwarding node

In the route discovery procedure, only the node within the request zone, called the forwarding node, can relay a route request packet to the next hop. Each node must have the ability to determine if it is a forwarding node. The determination of forwarding membership depends on the information of request zone carried in the request packet.

If a node receives the route request packet with TRIANGLEZONE information described in Section 4.2, it determines if it is inside the TRIANGLEZONE. If it is within the TRIANGLEZONE, then it is a forwarding node. There are many methods to determine if a node is within a TRIANGLEZONE. For example, in Fig. 4(a), when the node I_i receives route request from node S, it uses the TRIANGLEZONE information to determine that if angle γ is less than angle α and $l{*}\cos\gamma$ is less than $d + r$, then node I_i is a forwarding node. Otherwise, it is not a forwarding node. This judgment can be deduced by polar coordinates or coordinates transformation. We leave the development to the readers.

While a node receives a route request packet carrying RECTZONE information, it determines the forwarding node membership by checking if it is located in the RECTZONE. Considering the example in Fig. 4(b), when node I_i receives route request from node S, it uses the RECTZONE information to determine that if x_i and y_i is in the range of $[x_d - r, x_d + r]$ and $[y_d - r, y_d + r]$ respectively, then node I_i is a forwarding node. Otherwise, it is not a forwarding node.

When a node receives a route request packet including CIRCLEZONE information, the forwarding node membership can be determined by judging that if it is located in the CIRCLEZONE. In Fig. 4(c), when node I_i receives route request from node S, it uses the RECTZONE information to calculate the distance between itself and the destination.

If this distance is less than the radius of CIRCLEZONE (i.e., r), then node I_i is a forwarding node; otherwise, it is not a forwarding node.

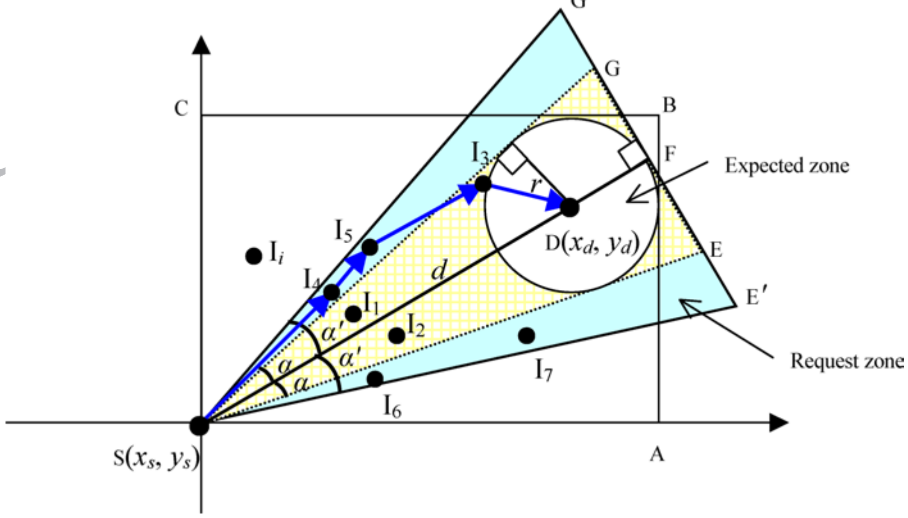
4.4 The policy of increase-exclusive search

Expanding the request zone to the entire network rapidly when route discovery fails, degrades performance and loses the benefits of a confined request zone based algorithm. We proposed an *increase-exclusive search approach* to improve this problem as follows. While a sender that used a TRIANGLEZONE in last routing process can not find the route to a destination within a timeout interval, it expands the TRIANGLEZONE by increasing the search angle with an angle increment and initiates a new route discovery until the search angle exceeds an angle threshold. Figure 5 shows TRIANGLEZONE $\Delta \mathrm{SE'G'}$ is expanded by increasing search angle from α to α' in the rerouting process. If a route is not discovered when the search angle exceeds a threshold, the source node expands the request zone to the entire network and rerouting again. In this case, the algorithm is similar to a traditional flooding algorithm. The more instances of rerouting, the longer latency in determining the route to destination. There exists a trade-off between latency of route construction and the routing overhead.

Inspecting Fig. 5 we found that the visited zone is included in a new expanded TRIANGLEZONE. It means that the route re-probing procedure will search the visited region repeatedly. Repeatedly searching for the same region causes a huge request message and exhausts a lot of resources of the network and forwarding nodes. The probability of finding a route in the visited zone by re-probing in a small time interval is very low unless the mobility is very high. Therefore, the visited nodes are not needed to take part in the rerouting process again unless it moves out. It means that a node forwarded a route request in previous route discovery procedure does not need to forward request message again in the route rediscovery. This is achieved by setting the sequence number for the request packet. We included a sequence number in the route request packet to indicate the freshness of this packet. If a forwarding node receives a request packet that has been seen, it discards this packet; otherwise, it forwards this packet to the next hop. It guarantees that a forwarding node forwards the route request with the same sequence number only once. If source node S defines the request zone to be a TRIANGLEZONE in the route rediscovery procedure, S initiates a route request packet with the same sequence number as the last route discovery procedure. We called this rerouting policy increase-exclusive search. Let's use the example in Fig. 5 to show how it works. In the first attempt, S uses a triangle ΔSEG as a request zone and initiates a route request with sequence number 1. There are three forwarding nodes, I_1, I_2 and I_3 in this TRIANGLEZONE. Assume that I_3 is out of the transmission range of node S, I_1, and I_2. Hence, only I_1 and I_2 can forward the request from S and record this event in their seen table. There is no other node can assist for relaying the request to the destination, so that the destination will not emit a route reply packet. Because S does not receive a route reply from destination D in the specified time interval, it claims that the first route discovery fails and retries a route discovery. If the last search angle α does not exceed the threshold, S enlarges the search angle to α' and resends a route request with the same sequence number (i.e., 1). In this time, when I_1 and I_2 receive the request, they discard it because they have forwarded a request with the same sequence number before. On the contrary, when I_3 receives the request, it forwards it to the next hop because it never forwarded this packet previously. The increase-exclusive search policy avoids huge routing traffic and collision, caused by the flooding policy used in LAR, when the routing process failed.

UNCORRECTED PROOF

Fig. 5 Increase-exclusive search process



On the other hand, the exclusive searching mechanism limits the number of route requests in the route rediscovery under an upper bound of $2n$, where n is the total number of nodes in the network. That is, the number of route request packets forwarded in the reroute procedure never exceeds twice the number of request packets relayed by flooding.

The increasing-exclusive search approach can enable intermediate nodes to avoid forwarding route re-discovery packets. But this traffic reduction mechanism may be harmful to route discovery since their neighbors may have changed during the discovery phases. The tradeoff between the routing overhead and the discovery efficiency should be addressed. The mobility problem is a challenge for mobile networks routing protocol design. It is hard to control and prevent completely. However, the re-probing interval of LARDAR is small so the effect for the increasing-exclusive search caused by the mobility problem is tolerable unless the mobility is extremely high. The unvisited nodes are allowed to forward a route request packet in the re-probing procedure so they are not affected by the mobility problem. On the contrary, the visited nodes are prohibited to forward route request that might affect the re-probing as follows. In the high mobility condition, a visited neighbor might move out the previous triangle-zone before next re-probing procedure. Because a visited node might has a chance to be an immediate node of the path to destination in a new triangle-zone, it is reasonable to allow this visited node to forward a route request packet in the rerouting process again. To this end, the visited flag in its seen table must be reset. To overcome this problem, the *seen table reset approach* is used to prevent the harmfulness of route discovery due to the movement of the visited neighbor. While the visited neighbor moves out the previous triangle-zone, the seen table reset approach is triggered to reset the related record in its seen table.

The selection of the value of angle increment and threshold is heuristic. It can be improved by genetic algorithm. To apply a genetic algorithm for a particular problem, we have to define or to select the following five components: (1) encoding schema for potential solutions. (2) initial population. (3) fitness evaluation function. (4) Alternating the composition of offspring. (5) Values for the various parameters that the genetic algorithm uses [30]. Here, we simply describe a simple algorithm for this problem as follows:

- *Encoding scheme:* The encoding process transforms the increase angle in angle space into eight bits string using binary coding. The angle increase range is set to [0, 90].
- *Fitness function:* The fitness function is defined in Eq. (5). The value of fitness function is set to 0, while there is no path can be found in the increase angle is selected to ?. If there is at least one path can be found while the increase angle is selected to ?, the value of fitness function is set to 1.

$$f(\theta) = \begin{cases} 0, \textit{ no path is found whlie increase angle is ?} \\ 1, \textit{ has path is found whlie increase angle is ?} \end{cases} \tag{5}$$

- *Objective function:* The objective function is defined to be the minimal path distance. Our goal is to find a path as shorter as possible. The angle of the triangle that finds a best path is the solution.

The genetic operation such as parent selection, crossover operation, and mutation operation will iterate a lot of times for problem solving.

4.5 Dynamic adaptation of request zone

In the DREAM algorithm, if an intermediate node has no location information of destination node, it can not forward the message to the next hop. The dependence on location information of all routing nodes increases the probability of routing failure. To improve this, we applied a *dynamic adaptation of request zone technique* in LARDAR. The location information of node includes *location* and *location information obtained time*. The *location information obtained time*, denoted T, is the timestamp when the location of node was collected. This timestamp is obtained according to the original node's local clock by itself. In order to update the location information of node, while a node sends a packet, it encapsulates its location, denoted L, and *location information obtained time* in this packet. Then, the location and timestamp, (L, T), are passed to other nodes by the packet. While a node received the packet, it extracts the location and timestamp carried in the packet and records both of them in routing table. Repeatedly, the location information of sender can be learned by all immediate nodes while a packet flows in the path. For example, while a node X wants to transmit a packet to another node Y, X will disperse its location information by encapsulating its location, L_X, and the timestamp when L_X was got, T_X, with this packet. While Y received the packet, it extracts the location and timestamp, (L_X, T_X), carried in the packet and records (L_X, T_X) in routing table. The location of node N is bound with the timestamp of the location is obtained in N, and then distribute to the network by learning. The *location information obtained time* is related to the clock of node that its location is concerned. Hence, the time synchronization mechanism among all nodes participating route discovery is not necessary.

In the route discovery phase, if the source node S has the location information of the destination node D in its routing table, it encapsulates this location information, (L_D, T_D), in route request packet for request zone adapting. When a forwarding node receives a route request, it checks that if

UNCORRECTED PROOF

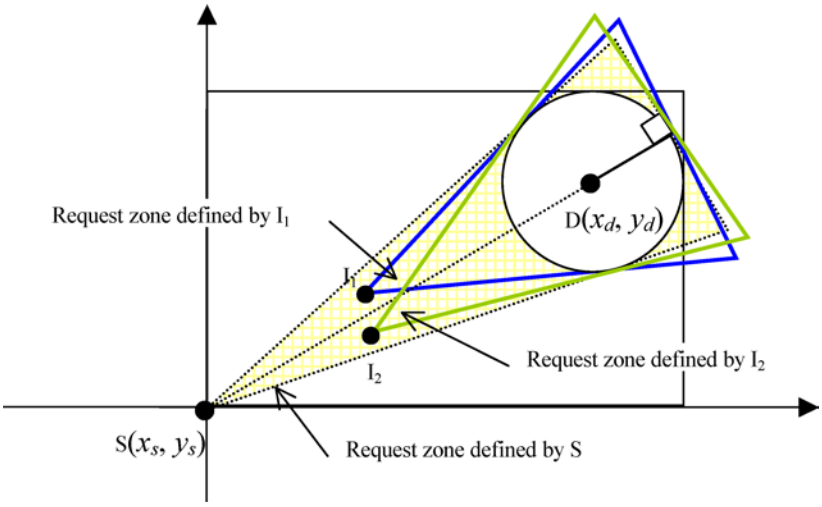


Fig. 6 Dynamic adaptation of request zone

there is location information of the recipient in the request packet or in its location database. If the location information of destination is found, it uses the freshest location information that is carried in the route request packet or stored in its location database to calculate the request zone dynamically (see Fig. 6). If the location database has newer location information of the destination, the newer location information must be included in the request packet and forwarded to the neighbor node. Alternatively, the fresher location information carried by the request will be used to update the location database. Each forwarding node repeats the same work to adapt the request zone and update the location database. If there is no location information in the request packet or in the location database of the forwarding node, it can not define a request zone. In this situation, a route request packet will be forwarded by flooding without the receiver's location included. The behavior of the dynamic adaptation of request zone approach is like a virtual directional or smart antenna that locks a target automatically. It is helpful to adapt the search angle and search direction precisely and define a request zone exactly. The success rate of route construction can be improved by this approach and the routing overhead can also be reduced. Another advantage of dynamic adaptation of request zone is to avoid selecting a next hop farther away from the destination node than the current node. That is, it prevents the request packet from forwarding in an opposite direction of the receiver. This is similar to the objective of the LAR scheme 2 [20].

4.6 The procedure of route discovery

In this subsection, we detail the procedure of route discovery in LARDAR as follows. When a source node S wants to transmit a message, it uses the location of the destination node to calculate an expected zone by the approach we described in Section 4.1. Next, it defines a request zone to include the expected zone according the definition we proposed in Section 4.2. Finally, source S sends a route request packet that includes the information of the request zone and the location information of the destination in the request packet. Then, node S waits for a route reply. When a node receives this packet, it uses the forwarding node membership determination rule (see Section 4.3) to determine if it is a forwarding node. If it is a forwarding node, it applies the dynamic adaptation of request zone policy (see Section 4.5) to calculate a new request zone and forwards this request packet accompany with the new request zone and the route from source node. The process is repeated until the destination is reached, if possible. When the destination node receives a route request packet, it unicasts a route reply packet along the reverse direction of the route that is recorded in the request packet to node S. The destination node includes its location information, velocity, location information obtained time and a copy of the route that is carried in the route request packet. The intermediate nodes, in turn, do the same, relaying the route reply by the path information within the route reply until the source node S is reached. If the source node S receives a route reply in the default time interval, it records the destination's location information, velocity, location information obtained time and the searched path that carried in the route reply into its routing table. If the route discovery is timeout, S will initiate a new route discovery by an increase-exclusive mechanism discussed in Section 4.4. After a route is constructed, the transmission of DATA packet starts.

4.7 Route recovery

If a route failure is detected by the node in the path to destination, it must recover the route as soon as possible. There are some alternatives of route recovery for the node that detected the route broken. The first alternative is the broken node sends a route error packet to inform the source node a

UNCORRECTED PROOF

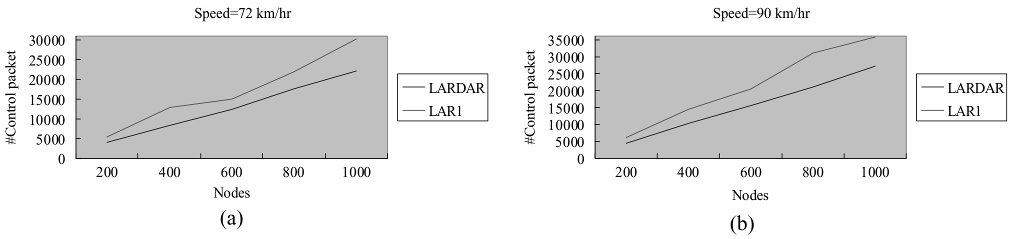


Fig. 7 Routing overhead (a) Speed = 72 km/hr, and (b) Speed = 90 km/hr

route failure has occurred. After having received a route error packet, the source node re-initiates a route discovery procedure to search a new path. Another alternative is to initiate a route discovery process by the broken node, called local search in LAR, to repair the broken path. This local search method reduces the overhead of route recovery as well as the latency of the route rediscovery. While the local search failed, it does route recovery by the first alternative.

5 Simulation results

We developed a simulator for our routing protocol LARDAR. The simulator was implemented within Global Mobile Simulation (GloMoSim) library by C++ language [27]. The GloMoSim library is a scalable simulation environment for mobile wireless network using parallel discrete-event simulation capability provided by PARSEC [28]. We tried to compare the performance of LARDAR with LAR scheme 1 (LAR1) that was implemented by J. Hsu and S. J. Lee and included within GloMoSim 2.03. The implementation of LAR scheme 1 followed the specification proposed in [20]. Other details are based on the discussions with Y. B. Ko. The route setup time, average hop count, data delivery rate, control packet (i.e., routing overhead) and collision for different network size is investigated by simulation.

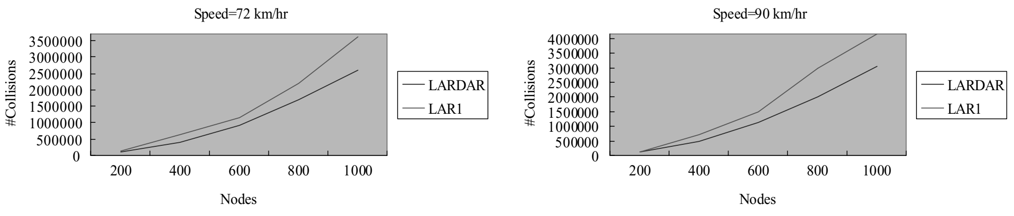
In our simulation, all network nodes were located in a physical area of size $1000 \times 1000\ \mathrm{m}^2$ to simulate actual mobile ad hoc networks. The network size was in the range of [200, 400, 600, 800, 1000] nodes that were generated according to a uniform distribution. The mobility model selected was the Random Waypoint model. For random waypoint, a node randomly selects a destination from the physical terrain, and then it moves in the direction of the destination in a speed uniformly chosen between the minimum and maximum roaming speed. After it reaches its destination, the node stays there for a specified pause time period. In our simulation, the value of minimum roaming speed was set to 0 km/hr. We considered two kinds of maximum mobility speeds, 72 km/hr and 90 km/hr. The pause time was fixed to 30 seconds. The propagation path loss model used in our experiment was the TWO-RAY model that uses free space path loss (2.0, 0.0) for near sight and plane earth path loss (4.0, 0.0) for far sight. The antenna height was hard-coded in the model (1.5 m). The radio frequency of each mobile node was 2.4 GHz. The radio bandwidth of each mobile node was 2 Mbps.

The simulation time of each run lasted for 400 seconds. Each simulation result was obtained from an average of the all simulation statistics. In each run, there are four application connections. The traffic generators used by the four application connections are CBR. The CBR simulates a constant bit rate traffic generator. The generators initiated the first packet (i.e., start time) in different time and sent a 512 bytes packet each time. The search angle increment and threshold used in LARDAR was 10° and 90° respectively.

Figure 7(a) and (b) show the distribution of routing overhead for different network size at two kinds of speed, 72 km/hr and 90 km/hr, respectively. The routing overhead was calculated as the total number of control packets transmitted in the route discovery procedure. The control packets included the route request packet and route reply packet for LARADR and LAR1. The number of control packets of both routing protocols increased when the network size enlarged. With a higher number of nodes, the density of node within the request zone increased, so the routing overhead also increased. The simulation result shows that LARDAR always had a lower routing overhead than LAR1. Because LAR1 defines a larger request zone than LARDAR and expands the request zone to the entire network rapidly if last route discovery procedure fails, it induces a higher routing overhead. As can be seen, LAR1 produced a larger amount of control packets that caused a higher probability of collision than is shown in Fig. 8. It also increased the route setup time and degraded the performance of data packet transmission (see Figs. 9 and 10).

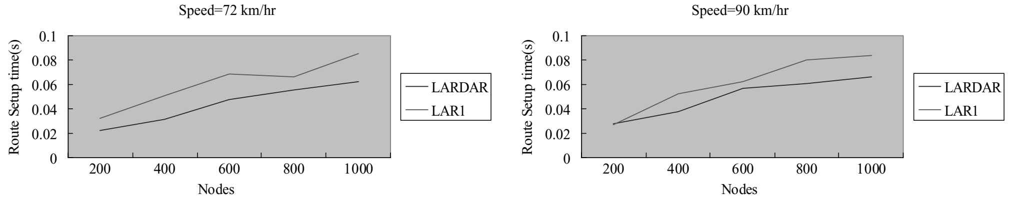
The total times of collision took place in LARDAR and LAR1 for different network size is shown in Fig. 8. The occurrence of collision for both routing protocols is raised with the network size. The number of collision occurred in LARDAR is much less than LAR1 for different roaming speeds. The probability of collision is proportional to the number of packets to be transmitted. The more nodes needed to transmit packets will produce a mass scale of traffic and cause more collision. As above, the request zone defined by LAR1 is larger than that of LARDAR so that a greater amount of nodes takes part in the route probing. The more

UNCORRECTED PROOF



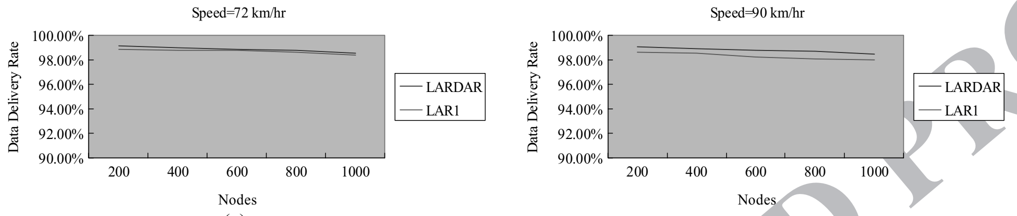
(a) (b)

Fig. 8 Times of collision (a) Speed = 72 km/hr, and (b) Speed = 90 km/hr



(a) (b)

Fig. 9 Route setup time (a) Speed = 72 km/hr, and (b) Speed = 90 km/hr



(a) (b)

Fig. 10 Data delivery rate (a) Speed = 72 km/hr, and (b) Speed = 90 km/hr

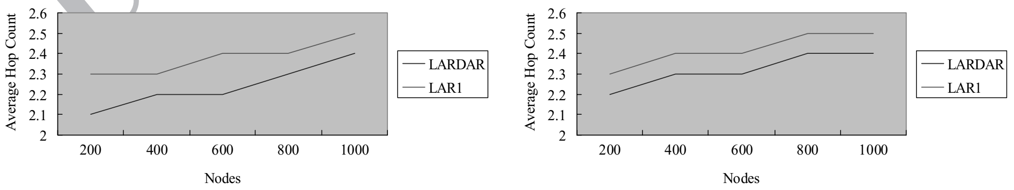
forwarding nodes participate in the routing process, the more control packets are broadcasted. This characteristic resulted in a higher chance of collision in LAR1 algorithm.

Figure 9 shows that the average time, called route setup time, required to construct a path to a destination node for LARDAR and LAR1 algorithm. Figure 9(a) and (b) show the average of route setup time simulated at speed 72 km/hr and speed 90 km/hr respectively. The average route setup time for both routing protocols at different speeds all increased when the network size was growing. LAR1 requires longer route setup time than LARDAR for different network sizes. The larger number of nodes take part in the route discovery will cause more routing traffic and occupy more network resources. The more number of packets be transmitted at the same time, the longer the delay and increased chance of collision. Collision induces packet retransmission and lengthens transmission time. This results in a longer route construction time.

The data delivery rate of LARDAR and LAR1 is illustrated in Fig. 10. In this figure, we can find that the data delivery rate of LARDAR is slightly higher than LAR1. All the delivery rates are over ninety-eight percent. The data delivery rate of LARDAR and LAR1 is decreased with the increase of network size for the increase collision.

Figure 11 shows the relationship between the average hop count for LARDAR and LAR1. The path length is increased

UNCORRECTED PROOF



(a) (b)

Fig. 11 Average hop count (a) Speed = 72 km/hr, and (b) Speed = 90 km/hr with the network size. LARDAR always built a shorter route than LAR1. As discussed earlier, TRIANGLEZONE restrains route request message to forward along a narrower space. It means that the request is forced to propagate in as straight a direction as possible. This is preferable in providing a higher chance to select a shorter route.

6 Conclusions

Using the location information of mobile nodes to assist routing can confine the route searching space into a smaller estimated range. The smaller space to be searched, the less routing overhead and broadcast storm problems will occur. In this paper, we proposed a location-based routing algorithm, called LARDAR, for mobile ad hoc networks. We improved the routing performance by three approaches. We defined a smaller triangle request zone to cover the possible destination positions. To adapt the estimated request zone and reduce the searching range, we applied a dynamic adaptation of request zone technique to redefine a more precise request zone. Finally, an increasing-exclusive search approach was used to redo the route discovery when the previous route discovery failed. It guaranteed that the areas of route rediscovery will never exceed twice the entire network. The comparison of our algorithm and LAR was studied through extensive simulation. The experiment results show that LARDAR outperforms LAR in many metrics, e.g., routing overhead, collision, route setup time, and average hop count. In the aspect of energy consumption, the above metrics are all very important for power-saving. Therefore, LARDAR can save more power and lengthen system lifetime than LAR.

Selecting a feasible value of angle increment and threshold had a great effect on the performance of our protocol. We described an angle increment and threshold selection algorithm by genetic algorithm. According to the intelligent decision capability of genetic algorithm we can get an optimal value of angle increment and threshold to improve the routing algorithm's performance.

References

1. C.E. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, Computer Communication. Rev. (October 1994) 234–44.
2. D. Johnson and D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: *Mobile Computing*, eds T. Imielinski and H. Korth. Kluwere Academic Publishers (1996).
3. R. Dube et al., Signal stability based adaptive routing (SSA) for ad-hoc mobile networks, IEEE Personal Communication (February 1997) 36–45.
4. D. Johnson, D.A. Maltz and J. Broch, The dynamic source routing protocol for mobile ad hoc networks (Internet-Draft), (December 1998).
5. C.-C. Chiang, M. Gerla and L. Zhang, Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks, Baltzer Cluster Computing 1(2) (1998) 187–196.
6. C.E. Perkins and E.M. Royer, Ad-hoc on-demand distance vector routing, in: *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. And Apps.* (February 1999) pp. 90–100.
7. E.M. Royer and C.E. Perkins, Multicast ad-hoc ondemand distance vector (MAODV) routing, IETF Internet Draft, draft-ietf-manet-maodv-00.txt (July 2000).
8. S.-J. Lee, W. Su and M. Gerla, On-demand multicast routing protocol (ODMRP) for ad hoc networks, Internet Draft, draft-ietf-manet-odmrp-02.txt (January 2000).
9. Educational observatory institute GPS page, http://www.edu-observatory.org/gps/gps.html
10. E.D. Kaplan, *Understanding the GPS: Principles and Applications, Artech House* (Boston, MA, February 1996).
11. G. Dommety and R. Jain, Potential networking applications of global positioning systems (GPS), Tech. Rep. TR-24, CS Dept., The Ohio State University (April 1996).
12. S. Capkun, M. Hamdi and J. Hubaux, Gps-free positioning in mobile ad hoc networks, in: *Proc. Hawaii Int'l. Conf. System Sciences* (January 2001).
13. J. Hightower and G. Borriello, Location systems for ubiquitous computing, Computer 34(8) (August 2001) 57–66.
14. K. Pahlavan, X. Li and J.P. Makela, Indoor geolocation science and technology, IEEE Communication Magazine (February 2002) 112–118.
15. M. Spreitzer and M. Theimer, Providing location information in a ubiquitous computing environment, in: *Proc. Symposium on Operating System Principles* (December 1993).
16. T. Imielinski and J.C. Navas, GPS-based addressing and routing, Tech. Rep. LCSR-TR-262, CS Dept., Rutgers University (March (updated August) 1996).
17. J.C. Navas and T. Imelinski, Geocast-geographic addressing and routing, *Proc. ACM/IEEE MOBICOM '97*, 3 (1997) 66–76.
18. C.-K. Toh, A novel distributed routing protocol to support ad-hoc mobile computing, Wireless Personal Communication (January 1997).
19. S. Basagni et al., A distance routing effect algorithm for mobility (DREAM), in: *Proc. 4th Annual ACM/IEEE Int. Conf. Mobile Computing and Networking, MOBICOM '98*, Dallas, TX, USA (1998) pp. 76–84.
20. Y. Ko and N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: *Proc. MOBICOM '98* (August 1998) pp. 66–75.
21. M. Joa-Ng and I.-T. Lu., A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks, IEEE Journal on Selected Areas in Communications 17(8) (1999) 1415–1425.
22. Y. Ko and N.H. Vaidya, Geocasting in mobile ad hoc networks: Location-based multicast algorithms, WMCSA'99 (February 1999).
23. L. Blazevic, S. Giordano and J. Boudec, Self-organizing wide-area routing, in: *Proc. SCI 2000/ISAS 2000* (July 2000).
24. B. Karp and H.T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in: *Proc. MOBICOM'00* (August 2000).
25. R. Jain, A. Puri and R. Sengupta, Geographical routing using partial information for wireless ad hoc networks, IEEE Personal Communications (February 2001) 48–57.
26. I. Stojmenovic, Position-based routing in ad hoc networks, IEEE Communication Magazine (July 2002).
27. L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia and M. Gerla, GlomoSim: A scalable network simulation environment, UCLA CSD Technical Report, #990027 (May 1999).
28. R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park and H. Song, Parsec: A parallel simulation environment for complex systems, Computer 31(10) (October 1998) 77–85.

UNCORRECTED PROOF

29. Y. Yu, R. Govindan and D. Estrin, Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, UCLA/CSD Technical Report, TR-01-0023 (May 2001).
30. M. Kantardzic, DATA MINING concepts, models, methods,and algorithms, IEEE Computer Society, Sponser (2003).

Tzay-Farn Shih received the B.S. degree in Information Management from Chinese Culture University, Taiwan, in 1992, the M.S. degree in Computer Science Engineering from Tatung University, Taiwan, in 1996, and the Ph.D. degree in Electrical Engineering from National Taiwan University, Taiwan, in 2006. He is presently an assistant professor of Computer Science and Information Engineering at Chaoyang University of Technology, where he initially joined in August 2006. He is currently an overseas member of the Institute of Electronics, Information and Communication Engineers (IEICE). His current research interests include computer simulation, computer networks routing protocol, wireless networks, Mobile Ad Hoc networks and sensor networks.

Hsu-Chun Yen was born in Taiwan, Republic of China, on May 29, 1958. He received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, in 1980, the M.S. degree in computer engineering from National Chiao-Tung University, Taiwan, in 1982, and the Ph.D. degree in computer science from the University of Texas at Austin, U.S.A., in 1986.

He is presently a Professor of Electrical Engineering at National Taiwan University, where he initially joined in August 1990. From August 1986 to July 1990, he was an Assistant Professor of Computer Science at Iowa State University, Ames, Iowa, U.S.A. His current research interests include Petri net theory, formal methods, design and analysis of algorithms, and complexity theory. Dr. Yen is an editor of International Journal of Foundations of Computer Science (IJFCS, World Scientific Publisher).

UNCORRECTED PROOF