VPDS: *V*irtual Home Region based *D*istributed *P*osition *S*ervice in Mobile Ad Hoc Networks

Xiaoxin Wu
Department of Computer Sciences
Purdue University
wu@cs.purdue.edu

***Abstract*— Position-based ad hoc routing algorithms have proved to have decent performance in delivery ratio and end-to-end delay. Position service is essential for such algorithms. In this paper, a distributed position service system, named VDPS, is proposed and evaluated. In the system, each node has a virtual home region (VHR), which is a geographic area. Nodes residing in a node's VHR will act as its position servers. A node updates its position to the servers, through which other nodes can obtain its position. To reduce the overhead on position management and simplify the operation, in VDPS, a VHR is divided into subregions. A region-based broadcast is used for position update, and a sequential searching is used for position retrieving. Different mechanisms for improving system robustness have been proposed. Mathematical models are built for analyzing system robustness and control overhead. Illustrated results show that VDPS can maintain a high system robustness at a relatively low control overhead.**

***Index Terms*— Distributed position service, position security, geographic routing, ad hoc networks**

I. Introduction

Position-based or geographic ad hoc routing algorithms, such as GPSR [1], have proved to have better routing performance than traditional ad hoc routing algorithms, such as AODV [2] and DSR [3], in end-to-end throughput and network scalability. Two prerequisites for position-based routing algorithms are: 1) each ad hoc node has to know its current position; 2) a source node has to know the updated position of its destination. The availability of the Global Positioning Service (GPS) [4] can help a node to calculate its own position. The more challenging research problem is how a source node can obtain the position of its destination.

One approach for obtaining a destination's position can be flooding. The source floods the position request in the network. When the destination receives the request, it replies the source with its most updated position. The approach is simple, yet the trade-off is the overwhelming control overhead especially in large-size networks. It is shown in [5] that flooding may cause a so-called broadcast storm problem, in which the ad hoc channel is occupied mostly for transmitting control messages. This undermines scalability, one of the major advantages of position-based routing algorithm.

Another approach is to build a position service system, in which any node can retrieve the position of another node from a position server (or servers). In an architecture where the ad hoc network is integrated with a fixed infrastructure such as a cellular-assisted ad hoc network [6], the position server can be attached to the fixed infrastructure, e.g., the cellular network. The integrated architecture makes position managements, including position update and position request/reply, less complex. However, in most cases, an ad hoc network works independently. Therefore, the position service system based on the ad hoc network itself, in which ad hoc nodes act as servers, needs to be designed.

The approach where the service system with one centralized position server in the ad hoc network is not practical, because the server may also be mobile so that it is difficult for a node to connect to the server. In addition, the server is the operation bottleneck for position management, as a server may only be a selected ad hoc node, which is not much more powerful than other normal ad hoc nodes. To address this concern, the position service system should be distributed among a number of servers deployed in the network. The research problems then are how a node updates its position (e.g., to which server a node should report its position) and how another can obtain this node's position (e.g., who is the right server to contact).

The general idea of a geographically based distributed position service system for mobile ad hoc networks is presented in [7]. Each node has a geographical region around a fixed center. The region is called a *virtual home region* (VHR). An ad hoc node updates its position information to all the nodes residing in its VHR, who are actually its position servers. The relationship between a node ID and its VHR center follows a hash function, so that other nodes can acquire this node's position by contacting the servers in the correct VHR.

In this paper, the idea of using VHR for position service is explored further. A VHR-based distributed position services system for ad hoc networks, named VDPS, is proposed. To lower the operating overhead on position management, only part of nodes will act as servers. To simplify the operation and reduce the overhead moreover, each VHR is divided into a number of subregions. A region-based broadcast scheme is designed for position update. A sequential searching method is used for position retrieving.

The major contributions of the paper are listed as follows:

- The detailed procedures for position management in the distributed position service system, including the routing between a node and the servers, are designed.

The work is supported by Institute of Information Infrastructure Protection (I3P) postdoc fellowship.

- The mechanisms for improving service robustness in the mobile environment are proposed.
- Analytical models are built, based on which the service robustness and the system operating overhead are evaluated.

The rest of the paper is organized as follows. In Section II the related works are presented. The details of the position service system is described in Section III. In Section IV, system robustness is analyzed. In Section V, management overhead such as the overhead on position update and retrieving are analyzed. Section VI shows the illustrated results. Section VII concludes the work.

II. Related Works

In [1], a greedy perimeter stateless routing protocol based on position information, called GPSR, is studied. The position information is exchanged locally among neighboring nodes by periodically sending out a beacon. Knowing the position of the destination, a source selects the neighbor who is geographically closest to the destination as the next hop and forwards the data packet to it. This next hop selects its own next hop following the same rule. The process is repeated until the packet reaches the destination. Such an approach is scalable since it does not need routing discovery and maintenance. In addition, it generally discovers the routes with the least hops. In the network with a fixed data rate, such routes result in higher end-to-end throughput and lower delay. GPSR may not always find the optimum route. When nodes are not uniformly distributed in the network, there will be dead ends, in which a node can not find any next hop closer to the destination. To solve this problem, the Face Routing and GFG (Greedy-Face-Greedy) schemes are proposed in [8]. Some other delivery-guaranteed methods are based on the single-path strategy, where a route from the source to the destination is built by using position information [9] [10].

GPS is normally used for a node in the ad hoc network to obtain its geographic position. Approaches for obtaining position information without GPS are also studied. A position deduction approach in [11] describes a way to calculate the node position based on the local connectivity. In some other work [12] [13], a network coordination system is built based on the distance measurement between nodes by determining the time of arrival for exchanged signals.

Two typical distributed position service systems for mobile ad hoc networks are GLS (Grid Location Service) [14] and DLM (Distributed Location Management) [15]. In GLS, the area covered by the entire network is divided into a hierarchy of grids with squares of increasing size. In each level of the grids, a node is assigned with a equal number of position servers. These servers have the closest ID distance to this node's ID, compared with all the other ad hoc nodes in the same grid. On the other hand, each node is a server for a number of other nodes, and has their updated positions (i.e., grid). Once a node needs the position information of another node, among all the nodes for whom it has the position information, it selects the node whose ID is closest to the target node and forwards the request. The process is repeated until the request reaches a node who has the position of the destination. GLS has relatively large position management overhead. To select the proper sever, a node has to make a global search at the beginning. Once a server leaves a grid, a new server has to be selected, which triggers a new search in that grid.

To address the problem in GLS, DLM is proposed. As GLS, in DLM, the area covered by the network is divided into the hierarchy of grids. Different from GLS, for each node, its position servers are decided by whether the nodes reside in a certain area, i.e., only the node residing in some certain grids can be this node's servers. The positions of these grids where the position servers of a node are located are the hashed result from the node's ID, so that any other node who needs this node's position knows to which grid they should forward the position request to. Position server is selected by default, whenever it moves into a grid.

Both GLS and DLM make the assumption that every node knows the grid hierarchy. This assumption is reasonable only when the network has a fixed range. While considering the dynamic topology of the ad hoc network, it is difficult for every node to has the precise grid information, which is necessary for the correct function of grid-based position service system.

III. VDPS Position Management

A. VDPS Overview

It is assumed that each ad hoc node can calculate its own position, e.g., through a GPS system. It is assumed that the nodes are uniformly distributed and the node density is not too low. In this case, the routes between any two nodes are normally available, which is necessary for position update/retrieving management.

Each node, or a *user*, has a circle-shaped VHR centered at a fixed point. The center of the VHR is the result of a hash function using this node's ID and the center of the ad hoc network as the inputs. The network center is roughly estimated at the time the network is initialized, which is useful for reducing the possibility that a VHR is out of the network coverage. The size of a VHR is determined by the server density and the node mobility. The size has to be large enough so that the system can achieve the required service robustness. The relationship between the VHR size and the system robustness are studied in more details in IV-B.

A node who resides in a user's VHR and handles the position management related to this user is called a position *server*. When network node density is not low, if every node will function as a server when it is in a VHR, high operating overhead will be generated. In VDPS, a node function as a position server at a probability. This probability depends on the node density, the size of VHR, and the system robustness requirement. In the rest of the paper, for the sake of presentation, it assumed that a probability has been chosen, which results in an equivalent server density. Term "server density" is used directly for system description and analysis.

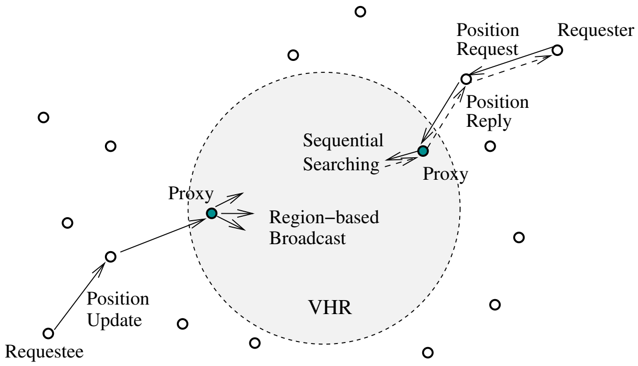

Fig. 1. VDPS position managements.

Other than users and servers, in this paper, a node that requests a position of another node is called a *requester*, and a user whose position is requested is called a *requestee*.

A user sends a *position update* message to the servers in its VHR. Once a node within the VHR receives the message, it will act as a *proxy*. Note that a proxy may not be a server. In VDPS, a VHR is further divided into subregions. The proxy will use the region-based broadcast to distribute the updated position information to all of the user's servers. More details about region-based broadcast are in III-B.

A requester sends a *position request* message to the requestee's VHR for its position. Once the message reaches a proxy within the VHR, the proxy will search the subregions sequentially until a server who has the requested position is found. The server then sends a *position reply* that carries the position information to the requester. More details about sequential searching are in III-C.

Since the position for the center of a VHR is known, the position update and position request messages are sent toward the VHR using position-based routing protocols, e.g., GPSR. To apply GPSR, neighboring nodes exchange positions with each other by sending "hello" messages.

The message flows and the operations for basic position managements are summarized in Fig. 1.

B. Position Update

In VDPS, position update is distance-based. A node updates its position when the distance from its current position to the position it reported to the server last time is over a threshold value. The distance threshold value is determined by the follows:

If the distance from a destination's current position and the position known to the source is smaller than this threshold value, when using position-based routing algorithms, the probability of a routing failure between the source and the destination caused by the out-of-date information is smaller than a required value.

The simulation results in [16] show that this threshold value can be approximately the half of the maximum ad hoc radio coverage. For example, if the ad hoc radio coverage is $250m$, the threshold value can be set as $125m$. Considering the pedestrian ad hoc users, the average time duration between two consecutive position updates is about tens of seconds. Such a low frequency of position update from a single node helps to reduce the control overhead for the position management and makes the system practical.

A user sends the position update message toward its VHR using GPSR where the center of the VHR is the destination. Based on its current position and the position of the VHR center, the user sends the position update to one of its neighboring node that is closest to the VHR center. If this neighboring node is still out of the VHR, it continues the GPSR packet delivery by finding its own next hop and forwarding the update message further. Once a node residing in the VHR receives the message, this node becomes the proxy.

The updated position information from a user should be stored by all of its servers. Since the radius of a VHR can be a few times as large as the maximum ad hoc radio coverage, to guarantee that every server in the VHR can receive the position update message, the proxy has to distribute the information.

A simple way for information distribution within the VHR is local flooding, i.e., each node within the VHR broadcasts the update message once. This leads to redundant transmissions and generates relatively large overhead. In VDPS, a region-based broadcast is proposed. As in a cellular network, a VHR is further divided into a number of hexagon-shaped subregions. The length of the diagonal of each hexagon has to be smaller than the maximum ad hoc radio coverage. Constructing the subregions in this way, a transmission from a node in a subregion can be received by all the other nodes in the same subregion. The transmission can also be received by nodes in some of its neighboring subregions, through which the position update can be broadcast in these subregions as well.

The number of subregions depends on the size of VHR, which is shown in Table I. Figure. 2 shows an example of dividing a VHR into subregions. If the radius of the VHR is $\sqrt{7}/2$ times as large as that of the ad hoc radio coverage, the VHR can be divided into seven hexagons, with the region number labeled at the right bottom. Here the diagonal of each hexagon is exactly the maximum ad hoc radio coverage.

TABLE I
SIZE OF VHR VS. NUMBER OF SUBREGIONS.

Size of VHR	Number of subregions
$R_{VHR}/r \leq 1/2$	1
$1/2 < R_{VHR}/r \leq \sqrt{7}/2$	7
$\sqrt{7}/2 < R_{VHR}/r \leq \sqrt{19}/2$	19
$\sqrt{19}/2 < R_{VHR}/r \leq \sqrt{37}/2$	37

Upon receiving a position update, the proxy adds the number of its own subregion to the message head and broadcasts the message. A node receiving the message finds out which subregion it belongs to, according to its own position and the position of the center of VHR carried in the position update message. If a node in a different subregion receives the message, it will back off for a random time and listen to

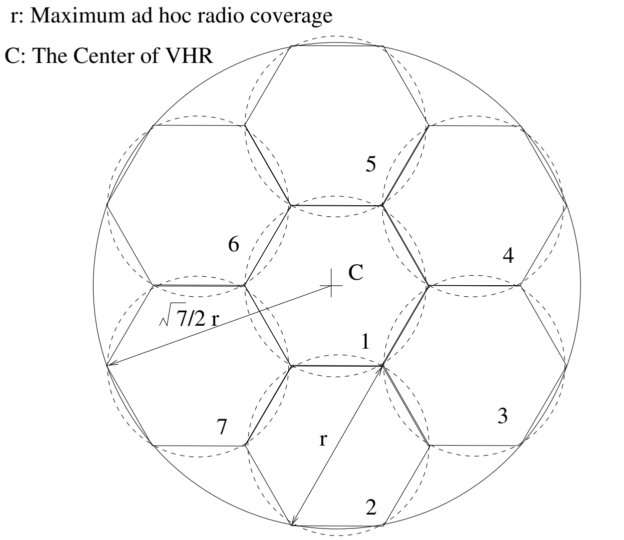

Fig. 2. Subregions of a VHR whose radius is $\sqrt{7}/2$ of the ad hoc radio coverage.

the channel. If no other node in its subregion broadcasts the message before it, it will add its own subregion number to the message and broadcast it. The position update will be broadcast in each subregion only once. The position information are finally distributed within the entire VHR after the position update message has been broadcast in all the subregions. In the example in Fig. 2, only 7 broadcasts are needed for distributing the updated position within the entire VHR.

C. Position Request/Position Reply

When a node requests a position, it sends the position request to the requestee's VHR. The position request also carries the updated position of the requester. This information is needed for a server to send the position reply to the requester. The request will be received by the proxy first. If the proxy happens to be a server that has the requested position, it sends back a position reply directly. Otherwise the proxy starts searching for the servers.

Since only one server who has the updated position of the requestee needs to be reached, the request is broadcast in the subregions sequentially until it is received by such a server. The process for searching a server is similar to the sequential paging process in the cellular network. More efficient sequential paging algorithms [17] can also be used to reduce the searching cost (overhead) and the searching delay. The position request needs to be delivered from one subregion to another during the searching process. Since the positions of the centers for subregions can be known, GPSR is used for the delivery.

When a server receives the request, it will judge whether it has the updated position of the requestee. A server has the updated position of a user if, since it receives the last position update from the user, it stays within the VHR all the time. If the request is received by a server who has the position information, or in other words, such a server is found, the server will send a position reply directly to the requester through GPSR.

A server judges whether it has the updated position upon receiving the position request. A server has the updated position of a user if, since it receives the last position update from the user, it stays within the VHR all the time.

The major procedures for sequential searching in the VHR are summarized as follows:

- A sender of the position request message, which is the proxy or a node that is included in the searching process, piggybacks its subregion number to message. It then selects a node in one of its un-searched neighboring subregions as its next hop and forwards the position request to it. This sender then waits for a time of t_w.
- Since the forwarding is equivalent to a broadcast of the position request in the sender's subregion, the forwarded request is also received by the nodes residing in the the sender's subregion. Among those if a server has the requested position information, it will wait for a random time between 0 and t_w. If it does not hear any transmission, it will reply to the sender an *ack*.
- If the sender receives an ack, it will then transmit a *finish* message. Upon receiving the message, the server who has the requested position will forward the information to the requester using GPSR protocol. On the other hand, the next hop of the sender will not process the message any further. The sequential searching is completed.
- If the sender senses a collision, i.e., at least two servers reply the ack simultaneously, it will transmit a *re-try* message. The servers who have sent out the ack in the previous attempt will contend to send the ack again. To avoid redundant transmission, the server who sends an ack will send out the position reply only after it receives the finish message.
- If the sender does not receive any ack after the time duration of t_w, it will transmit a *continue* message to its next hop in the neighboring subregion, which will repeat the searching process.
- If the requested position information is not available after all the subregions are searched, a *fail* message will be sent to the requester.

IV. System Robustness

The robustness of the position service system is determined by the probability of a successful position request. In this paper, a successful position request is defined as, when a requester sends a request to a VHR, it can reach a position server who has the updated position of the requestee. Note that a successful position request also relies on the successful deliveries for position updates and position requests. In this work, the research focus is on when the request reaches a VHR, how to improve the probability that there is at least one server who has the requested position.

In this section, different approaches that can be used to improve system robustness are proposed. Analytical models are built for robustness analysis in the proposed approaches.

A. Approaches for Robustness Improvement

The system robustness can be improved by increasing the number of position servers in a VHR. When there are more servers in the VHR, it is more likely that as a position request arrives, there are position servers who have the requested position information. The number of servers within a VHR depends on the node density and the probability that a node will function as a position server. Since node density is normally determined by the applications, system robustness can be improved by increasing the probability that a node can be a server.

System robustness can also be improved by increasing the size of VHR. As the size of a VHR increases, more servers may reside in it. In addition, the average dwelling time for a server in a VHR increases, and as shown later in the section, the longer dwelling time enhances the probability that when a position request reaches this server, it has the updated position of the requestee.

System robustness can also be improved by using a multi-VHR approach. For each node, its ID is hashed to different positions using different hash functions. Each position is the center of a VHR. Therefore, each node has multiple VHRs. These VHRs have different priorities and a pre-defined order, according to which a position update or a position request regarding to this node will be sent to. A user sends the position update message to the VHR with the highest priority first. If a server receives the message, it will reply an *ack* to the sender. If the user does not receive the *ack* because there is no server in that VHR or the position update can not be routed to the VHR, the user will send the position update to a VHR with a lower priority. This process is repeated until the updated position has been received by a server successfully. A request for the position of this user is sent to its VHRs in the same order as for position update, until an updated position is obtained. The multi-VHR approach is efficient in solving the problem that a node's VHR may be located in a blank area, where there are no ad hoc nodes deployed. The tradeoff is the increased operating complexity.

B. Robustness Analysis

It is assumed that a position update or a position request can always be routed to the targeted VHR. It is also assumed that the routing within the VHR is robust, i.e., once the message is received by a proxy, the following region-based broadcast or sequential searching can always be completed without routing problems. For mobility, nodes in the network are assumed to move randomly with the same average speed.

For a requester Q, it can obtain the updated position of its destination, U, only when both of the following conditions are satisfied:

1) At the moment the position request from Q reaches the VHR, there is at least one position server residing in the VHR.
2) This server has the updated position of U.

The second condition in the above is satisfied only when:

1) The server has at least received one position update from U since it enters the VHR.
2) The position update is before the arrival of Q's position request.
3) Between the position update and the position request, the server stays in the VHR all the time.

Based on the results in [18], the time that a randomly moved unit may stay in an area can be approximated as exponentially distributed with a mean time of $\bar{t}$, and

$$\bar{t} = \frac{\pi A}{E[v]L}. \tag{1}$$

Here A and L are the area and perimeter respectively, and $E[v]$ is the average speed of the mobile unit.

A node updates its position to its VHR when the distance between its current position and the position reported in its last update is more than a threshold value d_τ. The time between any two consecutive position updates from a node is then equal to the time that this node stays in a circular area with the radius of d_τ. Denote the time as t_u and its mean as $\bar{t_u}$, applying Eqn. (1),

$$\bar{t_u} = \frac{\pi d_\tau}{2E[v]}. \tag{2}$$

Similarly, the time that a position server stays in a VHR, denoted as t_s, is also exponentially distributed with a mean of $\bar{t_s}$. If the radius of a VHR is R_{VHR}, then:

$$\bar{t_s} = \frac{\pi R_{VHR}}{2E[v]}. \tag{3}$$

A sever X can reply Q with a updated position of U only if during the time period X stays in the VHR, denoted as $[0, t_s]$[1], X receives the position update from U at t_u and the position request from Q at t_r in a time order of $0 \le t_u \le t_r \le t_s$.

Given t_s, the probability of a successful position request, denoted as $p(t_s)$, is:

$$p(t_s) = p[0 \le t_u \le t_r \le t_s] = \int_0^{t_s} \int_0^{t_r} f_{t_u}(t_u) f_{t_r}(t_r) dt_u dt_r. \tag{4}$$

Here $f_{t_u}(t_u)$ is the probability density function of t_u, and $f_{t_r}(t_r)$ is the probability density function of t_r.

Due to the memoryless feature of exponential distribution,

$$f_{t_u}(t_u) = \frac{1}{\bar{t_u}} e^{-\frac{t_u}{\bar{t_u}}}. \tag{5}$$

During s' residence in the VHR, an arrival of a position request for u's position depends on how frequently its position is requested. This is determined by network traffic. For analysis tractability, we assume such a request can arrive at any time during s' residence with a equal probability, then:

$$f_{t_r}(t_r) = \frac{1}{t_s}. \tag{6}$$

Plugging Eqn. (5) and Eqn. (6) into Eqn. (4), integrated result can be obtained as:

$$p(t_s) = 1 - \frac{\bar{t_u}}{t_s}(1 - e^{-\frac{t_s}{\bar{t_u}}}). \tag{7}$$

[1]We initialize the time when Senters the VHR as 0 for analysis simplicity.

The average probability that a position request can be replied successfully when there is only *one* position server in the VHR, denoted as p, can be calculated by:

$$p = \int_0^\infty p(t_s) f_{t_s}(t_s) dt_s. \quad (8)$$

$f_{t_s}(t_s)$ is the probability density function of t_s and

$$f_{t_s}(t_s) = \frac{1}{\bar{t_s}} e^{-\frac{t_s}{\bar{t_s}}}. \quad (9)$$

p can be calculated numerically. When there are n position servers in the VHR, the probability for a successful position request, denoted as $P_{suss}(n)$, is:

$$P_{suss}(n) = 1 - (1-p)^n. \quad (10)$$

Based on the above results, the impact of both the server density and the area of a VHR on system robustness can be analyzed.

Assume a large number of N servers are uniformly distributed in an ad hoc network covering a large area of S. The server density, denoted as ρ, then is N/S. Assume a VHR covers an area of S_0, the probability that there are n servers in the VHR, denoted as $P_{ser}(n)$, is:

$$\begin{aligned} P_{ser}(n) &= \begin{pmatrix} N \\ n \end{pmatrix} (\frac{S_0}{S})^n (1 - \frac{S_0}{S})^{N-n} \quad (11) \\ &= \begin{pmatrix} N \\ n \end{pmatrix} (\frac{\frac{S_0}{S}}{1-\frac{S_0}{S}})^n (1 - \frac{S_0}{S})^N \\ &= \begin{pmatrix} N \\ n \end{pmatrix} (\frac{1}{\frac{S}{S_0} - 1})^n ((1 - \frac{S_0}{S})^{\frac{S}{S_0}})^{\frac{NS_0}{S}} \end{aligned}$$

Since $S >> S_0$, $\frac{S}{S_0} - 1 \approx \frac{S}{S_0}$, and $(1 - \frac{S_0}{S})^{\frac{S}{S_0}} \approx e^{-1}$. When $n << N$, $\frac{NS_0}{S} \approx \frac{(N-1)S_0}{S} \approx \frac{(N-2)S_0}{S} \cdots \approx \frac{(N-n)S_0}{S}$. Then:

$$\begin{aligned} P_{ser}(n) &\approx \frac{1}{n!} (\frac{NS_0}{S})^n e^{-\frac{NS_0}{S}} \quad (12) \\ &= \frac{1}{n!} (\rho S_0)^n e^{-\rho S_0}, \end{aligned}$$

which is a Poisson distribution. The probability of a successful position request in such a network, denoted as P, can be calculated based on Eqn (10) and Eqn (12), which is:

$$P = \sum_{i=1}^{\infty} P_{succ}(i) P_{ser}(i) \approx \sum_{i=1}^{I} P_{succ}(i) P_{ser}(i), \quad (13)$$

where I is a large number.

In the system where there are a number of N_V VHRs for each node, the probability of the successful position request, denoted as P_{multi}, is:

$$P_{multi} = 1 - (1-P)^{N_V}. \quad (14)$$

V. Control Overhead on Position Management

This section gives the analysis on the position management overhead. Especially, the overhead within the VHR, including both the overhead on position update and the overhead on position retrieving, is considered. The overhead involved in position update is further divided into *update overhead* and *broadcast overhead*. Update overhead is caused by the position update process in each server. The broadcast overhead is caused by position information distribution within the VHR. The overhead on position retrieving is mainly caused by the sequential searching.

A. Update Overhead

The update overhead is determined by how frequently a server receives position updates and stores the information. The major factors that affect the update load on a server are the user mobility, the user density, and the size of VHR.

It is assumed that the positions for the centers of VHRs for ad hoc users are uniformly distributed. Since each user has its own VHR, if the ad hoc users are distributed in the network with a node density of ρ_u, the density of the centers of VHRs is also ρ_u. For any position server, the number of VHRs it will reside in (i.e., the number of ad hoc users it will serve), denoted as n_u, is:

$$n_u = \pi R_{VHR}^2 \rho_u. \quad (15)$$

For an ad hoc user, it sends the position update when the distance between its current position and the position in its last position update gets greater than d_τ. The average time between any two consecutive position updates from the same user, t_u, can be referred to Eqn (2). Denote λ_u to be the frequency of position update from a user to its VHR, then $\lambda_u = 1/t_u$.

The frequency that a server receives a position update, denoted as f_c, is:

$$f_c = \lambda_u n_u = \frac{2E[V]\rho_u R_{VHR}^2}{d_\tau}. \quad (16)$$

B. Broadcast Overhead

Broadcast overhead depends on how frequently a node (not necessarily a server) in a VHR has to broadcast a position update message. If the flooding scheme is used, upon a position update, every node in the VHR has to broadcast the message once. The frequency of position update broadcast for an ad hoc node is the same as the frequency at which a server processes position updates, as shown in Eqn (16).

In VDPS, the region-based broadcast scheme is used. In such a scheme, for each position update, the number of broadcasts for the update message in the VHR is a fixed number equal to the number of subregions. Denote the number of the subregions in a VHR as N_R. Denote the probability that a node in the VHR has to broadcast the position update during a position update process as p_t. Then:

$$p_t = \frac{N_R}{\rho_u \pi R_{VHR}^2}. \quad (17)$$

The frequency of the position update in the VHR, f_c, has been calculated in Eqn. (16). The frequency for any node to broadcast the position update, denoted as f_b, is:

$$f_b = f_c p_t = \frac{2E[V]N_R}{\pi d_\tau}. \quad (18)$$

C. Overhead on Sequential Searching

A position request may have to be broadcast in a number of subregions before it reaches a server who has the requested position. The number of subregions that have to be searched for a successful position request is determined by how many times the position request will be broadcast before it reaches a subregion so that 1) a server (or more than one servers) is located in the subregion; and 2) the server(s) has the requested position. It is difficult to calculate the average number of searches directly. In this subsection, the analysis is simplified by making approximation.

Define $\bar{N}_s$ to be the average number of subregions that will be searched before a position request is received by any server. Note that the server may or may not have the requested position. Assume there are s servers in a VHR, and the VHR is divided into N_R subregions. It is also assumed that the sequential searching starts from a randomly selected subregion (depending on where the proxy is) and the subregions are searched in a random order. The probability that at least one server is found in the first subregion is $(1-(\frac{N_R-1}{N_R})^s)$. The probability that at least one server is found in the second subregion, which is on condition that no server is located in the first subregion, is $(\frac{N_R-1}{N_R})^s(1-(\frac{N_R-2}{N_R-1})^s)$. Similarly, the probability that at least one server will be found in the ith search, defined as $p\{n_r = i\}$, is:

$$\begin{aligned} p\{n_r = i\} &= (\frac{N_R-1}{N_R})^s \times (\frac{N_R-2}{N_R-1})^s \times \cdots \quad (19) \\ &\quad (\frac{N_R-(i-1)}{N_R-(i-1)+1})^s \times (1-(\frac{N_R-i}{N_R-i+1})^s) \\ &= (\frac{N_R-(i-1)}{N_R})^s - (\frac{N_R-i}{N_R})^s. \end{aligned}$$

$\bar{N}_s$ can then be calculated by:

$$\begin{aligned} \bar{N}_s &= \sum_{i=1}^{N_R} i p\{n_r = i\} \quad (20) \\ &= 1 + \sum_{i=1}^{N_R-1} (\frac{N_R-i)}{N_R})^s. \end{aligned}$$

Define $p_i(i)$ to be the probability that in a subregion, there are i severs. Then:

$$p_i(i) = \begin{pmatrix} s \\ i \end{pmatrix} (\frac{1}{N_R})^i (\frac{N_R-1}{N_R})^{s-i}. \quad (21)$$

Define p_x to be the probability that the requested position can be obtained when there are servers in any subregion. p_x can be calculated by:

$$p_x = \frac{\sum_{i=1}^{s} P_{succ}(i) p_i(i)}{1-p(0)}, \quad (22)$$

where $P_{succ}(i)$ can be obtained from Eqn. (10). The enumerator is the probability that there is at least one server in the subregion.

It is well known that if for a single attempt, the probability of success is p, then the average number of attempts for the success, defined as $\overline{n_{att}}$, is:

$$\overline{n_{att}} = \sum_{i=1}^{\infty} i(1-p)^{i-1} p = \frac{1}{p}. \quad (23)$$

The average number of times that a request has to be retransmitted before it reaches a server who has the requested position, defined as $\overline{N_{succ}}$, can then be approximated by:[2]

$$\overline{N_{succ}} \approx \frac{\bar{N}_s}{p_x}. \quad (24)$$

VI. Illustrated Results

Unless otherwise specified, each user has a single VHR. The distance threshold value above which a user has to send a position update, d_τ, is set as $125m$. The average speed of a mobile node, $E[v]$, is set as $3m/s$. The maximum ad hoc radio coverage r is $250m$. The size of a VHR is defined as R_{VHR}/d_τ, where R_{VHR} is the radius of the VHR.

Term *server density* is used for the sake of presentation, which is determined by ad hoc node density and the probability that a node acts as a position server.

A. System Robustness

Figure 3 shows how increasing the size of a VHR improves system robustness. The y-axis is the probability that upon the arrival of a position request, a server in the VHR has the updated position of the requested node, and consequently the position request is successful. The result shows the probability increases as the size of VHR increases. This is because the time that the server stays within a larger VHR is longer. When the frequency of position update is given [3], considering that a position request may arrive at any time during the server's dwelling time in the VHR while the arrival of position updates follows a Poisson process, as the dwelling time increases, it is more likely that the server has received at least one position update before it receives the position request. Figure 3 also shows that the size of VHR should not be too small, otherwise the probability for a successful position request will be low.

Figure 4 shows that the system robustness improves as the number of servers in a VHR increases. This number gets larger as the server density increases, or the size of a VHR increases. As shown in the figure, when the number of the servers increases, the position request is more likely to be successful because the probability that at least one of these servers has the requested position upon the arrival of a position request increases. Note that the increased size of VHR also results in the longer dwelling time of a server. The impact of server dwelling time has been shown in Fig. 3.

[2]This is only an approximate calculation because in sequential searching process, searches in different subregions are not independent.

[3]The frequency is determined by the position update distance threshold value and the node mobility.

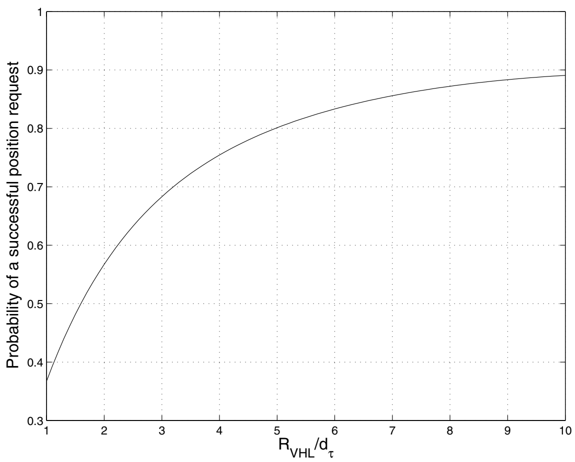

Fig. 3. System robustness vs. the size of VHR: the single server case.

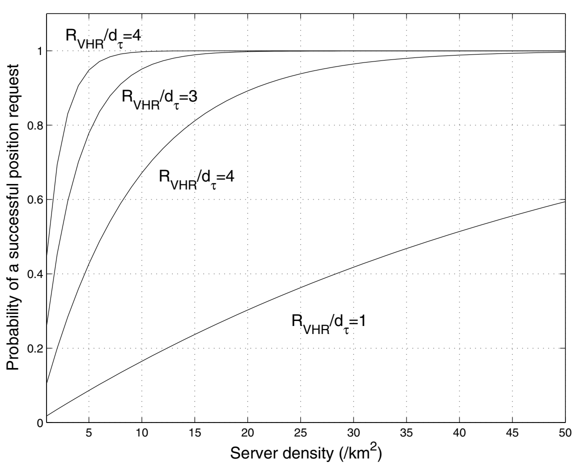

Fig. 4. System robustness vs. the number of servers in a VHR.

Figure 5 shows the system robustness in the approaches with different number of VHRs. The three curves on the top of the figure are the results from the schemes where the overall area covered by the VHR(s) is the same. In this case, given a server density, each user has the same number of servers. It is observed that the scheme with larger size of VHR (i.e., smaller number of VHRs) has the higher probability of a successful position request, since the server dwelling time in a VHR is longer. The curve at the bottom is the result from the scheme with single VHR, which is used for comparison.

Table II lists the probabilities of a successful position request in a single-VHR scheme at different node mobility. It is observed that the average node speed $E[V]$ has very little impact on the probability of a successful position request P. The reason is that the speed affects both the server dwelling time, which depends on the server speed, and the position update frequency, which depends on the user speed. When $E[V]$ gets higher, although the server dwelling time decreases, a node updates its position more frequently. Thus, the probability that upon receiving a position request, the server has the requested position does not drop significantly.

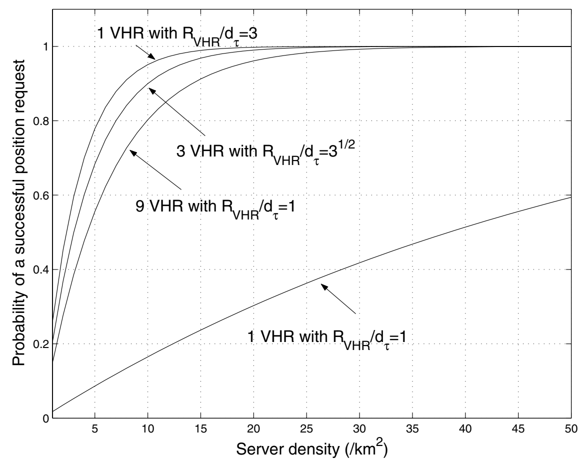

Fig. 5. System robustness in schemes with different number of VHRs.

TABLE II
IMPACT OF NODE MOBILITY ON SYSTEM ROBUSTNESS.

$E[V](m/s)$	3	9	15	21
P $(R_{VHR}/d_\tau = 2)$	0.8922	0.8920	0.8918	0.8916
P $(R_{VHR}/d_\tau = 3)$	0.9976	0.9976	0.9976	0.9976

B. Position Management Overhead

Figure 6 shows the broadcast overhead caused by position distribution in the VHR , i.e., how frequent a node has to broadcast a position update message upon the position update arrivals. The overhead in both region-based and flooding schemes are shown for comparison. It shows that the broadcast overhead in the region-based broadcast scheme is very small. When the size of VHR is fixed, the overhead is constant as the node density changes. On the other hand, the overhead in the flooding scheme increases as the density of ad hoc nodes increases, due to the increased number of position updates to a VHR. In the flooding scheme, the overhead also increases as the size of the VHR increases, since the enlarged VHR means a node will be included in an increased number of VHRs, so that it has to process more position update broadcasts. Not shown in the figure, the broadcast overhead in the region-based scheme also increases as the size of VHR increases due to the increased number of subregions. Note that the update overhead, i.e., the frequency for processing position update in each position server, is the same as the frequency for a node to retransmit the position updates in the flooding scheme, which depends on the node density and the size of VHR. If the size of the VHR is not too large, the number of position updates that a sever has to process is several times a second even when the ad hoc node density is high.

Figure 7 shows the impact of node mobility on overhead. Here the node density is $100/km^2$ and $R_{VHR}/d_\tau = 3$. Both the update overhead and the broadcast overhead increase as the node speed increases. This is obvious because the more mobile

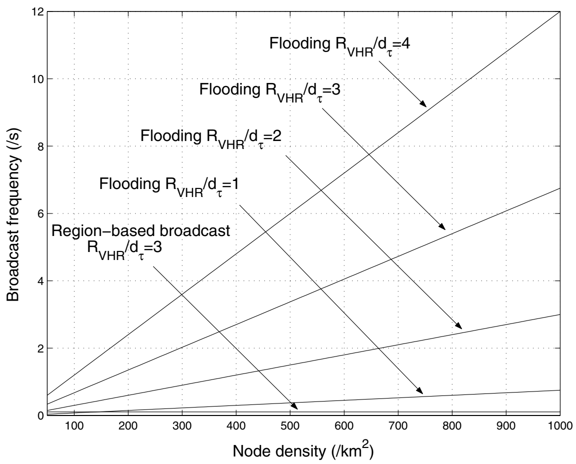

Fig. 6. Node broadcast frequency for position updates.

a node is, the more frequently it has to update its position, and more position update and broadcast are involved.

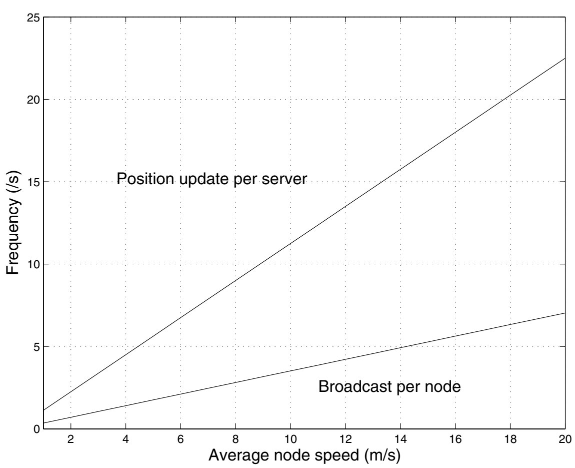

Fig. 7. Overhead vs node mobility.

Figure 8 shows the average number of subregions that have to be searched before a server has the requested position is reached. When the number of servers in a VHR increases, the number of searches decreases because a server can be reached after searching a smaller number of subregions. A larger-sized VHR leads to higher searching cost because a VHR has to be divided into more subregions.

To find out the update overhead at the network level, the frequency of the position update process in the area of a square kilometer where node densities ρ_u are different is shown in Fig. 9. Schemes with different VHR sizes are compared. The densities of the servers in these schemes are set to make the correspondent probabilities of a successful position update around 0.95. The figure shows that the frequency of position update process at the network level increases as the node density increases. However, the frequency decreases slightly when VHR size increases. This is because as the size increases, the number of the servers needed in the VHR to meet the

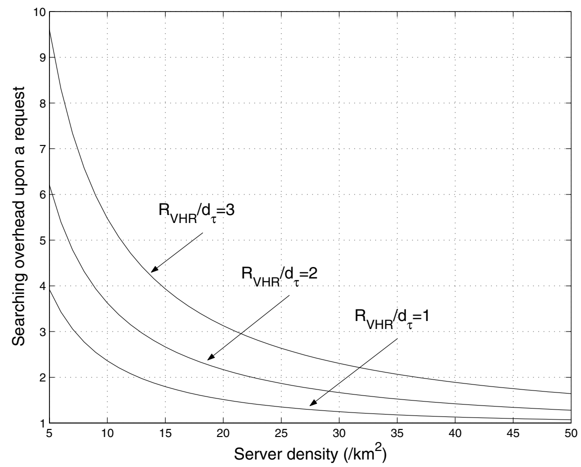

Fig. 8. Number of subregions that need to be searched for a successful position request.

requirement for the probability of a successful position request decreases. The reduced update overhead caused by the smaller number of servers in each square kilometer overwhelms the increased overhead at each server caused by the larger number of the nodes that a server has to serve.

In Fig. 10, we show the correspondent searching overhead under the robustness of 0.95 when VHRs with different sizes are used. Generally, the overhead increases as the size of VHR increases. This is because when a larger VHR is used, fewer servers are needed to meet the robustness requirement. More subregions need to be searched before the position request reaches a server. The impact of the increased number of subregions to be searched overwhelms the impact from the increased probability that a server has the requested position. There is a large increase for searching overhead at $R_{VHR}/r = 2.2$, where the number of subregions in a VHR changes from 19 to 37.

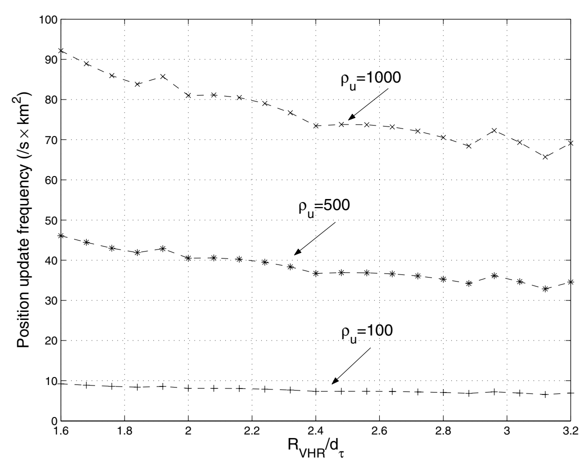

Fig. 9. Position update overhead at system robustness of 0.95.

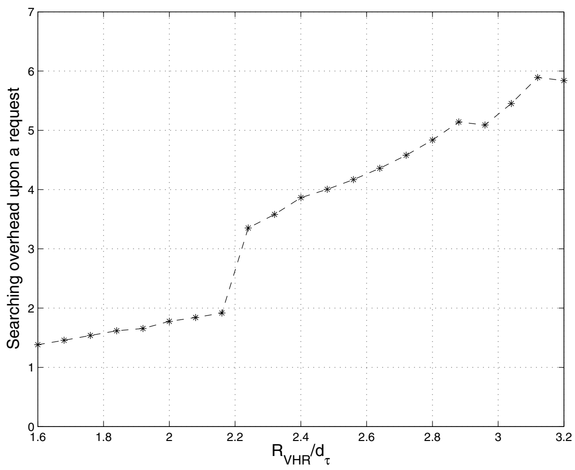

Fig. 10. Searching overhead at system robustness of 0.95.

VII. CONCLUSIONS

A distributed position service system for ad hoc networks, named VDPS, is proposed and evaluated. The design is VHR-based, while a VHR is further divided into subregions. Mechanisms on overhead reduction and robustness improvement are designed. The detailed procedures for position management in a VHR are given. Mathematical models are built for system robustness and overhead analysis. The illustrated results show that VDPS has decent system robustness while the overhead caused by information distribution is reasonable.

REFERENCES

[1] B. Karp and H. T. Kung, *GPSR: Greedy Perimeters Stateless Routing for Wireless Network*, in proceedings of MOBICOM'00, 2000.

[2] C.E. Perkins and E.M. Royer, *Ad-hoc On-Demand Distance Vector Routing*, in proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp.90–100, 1999.

[3] D. Johnson and D. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, in proceedings of ACM SIGCOMM-Computer Communications Review, 1996.

[4] B. Wellenhoff, H. Lichtenegger and J. Collins, *Global Positions System: Theory and Practice*, Fourth Edition. Springer Verlag, 1997.

[5] Y. Tseng, S. Ni, Y. Chen, and J.Sheu, *The Broadcast Storm Problem in a Mobile Ad hoc Network*, in proceedings of MOBICOM'99, August, 1999.

[6] B. Bhargava, X. Wu, Y. Lu, and W. Wang, *Integrating Heterogeneous Wireless Technologies: A Cellular-Assisted Mobile Ad hoc Networks*, Mobile Network and Applications, No. 9, 393-408, 2004.

[7] L. Blazevic, L. Buttyan, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec, *Self-Organization in Mobile Ad hoc networks: The Approach of Terminodes*, IEEE Personal Communications, pp. 166–174, June, 2000.

[8] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, *Routing with Guaranteed Delivery in Ad-Hoc Wireless Networks*, ACM Wireless Networks, pages 609–616, November 2001.

[9] R. Jain, A. Puri and R. Sengupta, *Geographical Routing Using Partial Information for Wireless Ad hoc Networks*, IEEE Personal Communication, February 2001, 48-57.

[10] I. Stojmenovic, M. Russell, and B. Vukojevic, *Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks*, IEEE Int. Conf. On Parallel Processing, pages 173-180, Aug. 21-24, Toronto.

[11] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, *Localization from Mere Connectivity*, in The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2003.

[12] S. Capkun, M. Hamdi, and J.-P. Hubaux, *GPS-free Positioning in Mobile Ad Hoc Networks*, Proceedings of 34th HICSS, Japan, 2001.

[13] A. Rao, A. Ratnasamy, C. Papadimitriou, S. Shenker, and L. Stoica, *Geographic Routing without Location Information*, Proceedings of MOBICOM'03, 2003.

[14] J. Li, John Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, *A Scalable Location Service for Geographic Ad Hoc Routing*, in ACM Mobicom 2000.

[15] Y. Xue, Ba. Li, and K. Nahrstedt, *A Scalable Location Management Scheme in Mobile Ad-hoc Networks*, in 26th Annual IEEE Conference on Local Computer Networks (LCN'01), 2001.

[16] X. Wu and B. Bhargava, *AO2P: Ad Hoc On-Demand Position-Based Private Routing*, submitted to IEEE Transaction on Mobile Computing.

[17] R. Rezaiifar and A. M. Makowski, *From Optimal Search Theory to Sequential Paging in Cellular Networks*, Selected Areas in Communications, IEEE Journal on , Volume: 15 , Issue: 7 , Sept. 1997 Pages:1253 - 1264.

[18] R. Thomas, H. Gilbert, and G. Mazziotto, *Influence of the Moving of the Mobile stations on the Performance of a Radio Cellular Network*, in proceedings of Third Nordic Seminar, 1988.