Service Discovery in Mobile Ad Hoc Networks Based on Grid

Hua-Wen Tsai, Tzung-Shi Chen, Member, IEEE, and Chih-Ping Chu, Member, IEEE

Abstract—Service discovery is a crucial operation for the usability of mobile ad hoc networks (MANETs). Traditional solutions to service discovery that were adopted on the Internet are not well suited for MANETs because of their dynamic nature. This paper presents a novel service discovery protocol based on a hierarchical grid architecture in an ad hoc network. The geographical area of a MANET is divided into a 2-D logical hierarchical grid. The proposed scheme registers the information of available services to a specific location along a predefined trajectory. To enhance resource availability and effective discovery, each grid cell selects a directory for caching available services. This paper utilizes the transmitting trajectory to improve the efficiency of registration and discovery. First, the service provider registers a service along the proposed register trajectory. The requestor then discovers the service along the discovery trajectory to acquire the service information. This paper also proposes an improved process to avoid the sparse node network topology. This paper adopts analysis and simulation to measure the protocol performance. The analytical and simulation results show that the proposed protocol can raise the discovery success ratio and reduce the discovery cost more than the existing geographical service discovery protocol.

Index Terms—Grid, mobile ad hoc networks (MANETs), mobile computing, service discovery, wireless networks.

I. INTRODUCTION

A self-organizing dynamically reconfigurable wireless network without a fixed infrastructure or central management, mobile ad hoc networks (MANETs) are extensively applied in various areas, including military or postdisaster rescue operations, temporary group collaborations at conferences or lectures, and sensor networks [1]. MANETs have no fixed infrastructure support; therefore, wireless nodes automatically form cooperative infrastructures, i.e., the nodes agree to relay packets for each other. Wireless nodes have basic communication capability, including routing, address allocation, name resolution, and service discovery. Additionally, wireless nodes have mobility capability so that they can freely join or leave a MANET. Hence, the network topology of a MANET frequently changes. However, traditional service discovery solutions, which are adopted in the Internet, are unsuited to MANETs because of their dynamic nature.

Interconnection systems, e.g., routing protocols, have been developed in recent years [2]–[5]. Recent studies [6]–[9] have considered service discovery in terms of applications. The network has to supply service discovery, whereas a client wants to access a service provided by a server. Some of service discovery scenarios for MANETs are described as follows.

In MANETs, a network connection can be regarded as a service [10]. Only access points can provide such a gateway service to the members of MANETs. The requestor has to acquire this service using service discovery.

In a mobile gaming environment, a player holds a wireless handheld device to join a game session. Consequently, the game host is also viewed as a service. Additionally, collaborative applications are also similar to such game services.

For intelligent transportation systems [11], service discovery can assist users to acquire real-time trip tip and traffic data, including the traffic block, congestion information, accident information, and density information of streets, from a real-time information system. Additionally, wireless devices will form a part of basic equipment in cars in the near future. Service discovery between vehicles is also supported by server- and client-based technologies.

In electronic parking systems, service discovery can assist the driver in discovering the available parking place as a car enters a parking lot. The parking place is regarded as a service.

Service discovery is also applied to commerce. For example, sellers provide information about goods for customers, who can obtain the information about goods of interest to them using service discovery.

Several major challenges [9] of service discovery systems for MANETs are derived from these scenarios.

The network topology of a MANET is nonstatic, and the node position can be changed at any time. Hence, a good service discovery protocol can accept intermittent connectivity.

Mobile devices, in general, have functional limitations, including a smaller radio propagation range, bandwidth limitations, battery power consumption, and memory capacity. To fully utilize equipment resources, a service discovery protocol aims to minimize traffic and resource consumption.

MANETs are typically applied to small areas, e.g., battlefields and disaster rescue sites. With the popularity of wireless devices, MANETs are also applied to public places, including colloquia, botanical gardens, exhibitions, and metropolitan areas.
areas. Designing an effective service discovery protocol for large MANETs is a challenge.

Several service discovery protocols [12]–[16] are currently being developed and standardized by different industry consortiums and organizations. However, these systems were not designed for dynamic infrastructureless networks and, thus, are not appropriate for MANETs. This paper proposes an efficient service discovery protocol based on a grid architecture for MANETs called SGrid. The geographical area of a MANET is treated as a 2-D logical grid [17]–[19]. This work proposes a hierarchical mechanism for identifying its grid level. One node in its grid cell is elected as a directory for caching available service descriptions. This paper adopts transmission trajectory to raise the efficiency of discovery protocols. One provider registers its services to a predefined trajectory to share services for other nodes. The trajectory crosses the network center, and the directories on the trajectory have to cache the service descriptions. One requestor sends the register/discovery packets along the predefined directions to discover service toward the center line of network while it needs a service. Analytical and simulation results show that SGrid effectively raises the discovery success ratio and lowers the discovery cost. This paper also solves the empty area problem for sparse node networks.

The rest of this paper is organized as follows. Section II discusses the related work. Section III then describes the grid-based network model and the overview of the proposed protocol. The algorithm is described in detail in Section IV. Section V presents the results of performance analyses and simulations. Conclusions are drawn in Section VI.

II. RELATED WORK

This section presents the existing service discovery protocols. Some conventionally adopted protocols, including Sun’s Jini [12], the service directory service Universal Description, Discovery, and Integration [15], and IBM’s Salutation [14], rely on a central directory to register and discover services. The information of available services is concentrated in directories. The route and service advertisement are based on multicasting, and the service register is based on unicasting. Centralized directory discovery is, in general, appropriate for Internet or wireless infrastructure-based networks, but such an architecture is inappropriate for MANETs because of their dynamic nature. Furthermore, the centralized directory easily suffers from routing bottlenecks. Although this architecture can create additional directories to solve the scalability problem, the construction cost rises with the number of directories.

Some distributed approaches, including Microsoft’s Universal Plug and Play [16], the IETF’s Service Location Protocol [13], and Bluetooth’s Service Discovery Protocol, have been developed to reduce the overload of the centralized directory. Such approaches are typically based on either pull- or push-based broadcasting. However, their major disadvantage is the frequent broadcast obstruction on the network. To alleviate this problem on MANETs, the distributed service discovery protocol [20] has been developed to register the information of available services to a virtual backbone based on clustering.

Fig. 1. Example for the register and discovery of GCLP.

The cluster head acts as a directory for caching the available service information. A virtual backbone is formed between these clusters. To register/discover services, every node constructs a multicast tree to link with the virtual backbone. In this protocol, every node needs to maintain cluster and gateway information because of node mobility. Therefore, extra overhead is generated in maintaining the cluster and multicast tree.

One geographical routing approach, which is called the Geography-based Content Location Protocol (GCLP) [21], has recently been proposed for service discovery. GCLP applies the physical location information to lower the amount of proactive traffic. The service register and discovery processes should not rely on broadcasting. The register/discovery occurs along a crisscross trajectory with four directions: 1) east; 2) west; 3) north; and 4) south. The provider selects four relay nodes that are the farthest node in each direction and are responsible for forwarding the register packets. The register packet is transmitted in all four directions until no relay node is available. The register is broadcast along the crisscross trajectory, and therefore, a node caches the service description when it receives the register packet. That is, all the nodes near the crisscross trajectory are the directories. The service description consists of the service type, provider ID, and provider location. The requestor discovers a service along the discovery trajectory that is the same as the register trajectory. The requestor can discover the service information from directories, whereas the discovery trajectory meets the register trajectory.

Fig. 1(a) illustrates an example for GCLP. Provider P registers a service along the crisscross trajectory. All nodes near the trajectory are the directories. Requestor Q can get the provider information, because it is a directory for P. Another requestor R does not know the service information, and therefore, it discovers a service along the discovery trajectory. Receiving the discovery packet, several directories, called response directories, respond to the requestor with the service information. R obtains the information from the west and the north trajectories. This scheme efficiently decreases the cost of discovery and register, making it suitable for large networks.

However, this solution has two major drawbacks. First, it assumes that the node quantity is sufficient to cover the complete network. This method cannot be assured to work well in a sparse network. The east trajectory of P cannot traverse the
empty area, as indicated in Fig. 1(b), causing the register to break off in the middle of the east register trajectory. In this situation, the requestor R still discovers the service information from the west discovery trajectory. However, the requestor Q cannot discover the service because of the empty area. Second, there exists heavy storage and communication consumption. In GCLP, nodes have to cache the service descriptions when they hear the register packets so that all the nodes that are close to the discovery trajectory are directories, as in Fig. 1(a). The register storage consumes too many resources. In addition, the discovery response of GCLP causes too much communication consumption, because the requestor receives the same information from many different directories that receive the discovery packet and respond to the requestor with the service information. Therefore, R receives the same packets from 15 directories, as in Fig. 1(a).

Another grid-based solution [7] provides position discovery for mobile nodes. Every node periodically updates its current position to a set of directories without knowing the actual identity of directories. The position update and query are assisted by a predefined node sequence and geographical hierarchy. Another solution, i.e., Service Rings [22], establishes a hierarchical service ring architecture. Every ring has a designated directory for caching the service information. The directories are also linked to form high-level rings. Service discovery is performed along the ring level. However, routing maintenance in the ring is expensive, particularly in a large network. The authors of Service Rings also proposed Lanes [23], which adopted a completely logical overlay structure. To solve problems about the lack of fixed commonly known nodes and relatively extensive ping messages in MANETs, Lanes adopts anycast routing to send messages from one lane to another. However, the communication between two neighboring lanes may cross the entire network.

Many approaches have been presented for service discovery. The advantages and disadvantages of different solutions are listed as follows. Centralized discovery reduces the discovery cost but requires the directory information to be known by other nodes. Additionally, the service information is centralized in a group of directories, thus ensuring that the service information is correct. However, centralized discovery is not suitable for MANETs because of their dynamic nature. Distributed discovery protocols enable every node to act as the directory for caching the available service descriptions. Although distributed discovery is more appropriate than centralized discovery for MANETs, its broadcast strategy incurs heavy traffic. GLCP adopts geographical information to lower the register and discovery cost but is only suitable for high node density in MANETs. It can cause heavy storage and communication consumption. This study aims at reducing the shortcomings of the existing methods, raise the discovery success ratio, and lower the resource consumption. This study proposes a grid-based service discovery protocol that integrates the benefits of the centralized directory and geographical discovery. Although every node has to equip the device that supports geographical location, including GPS or other location-aware devices, this technology will be popularized in the near future.

III. PROTOCOL OVERVIEW AND NETWORK MODEL

This section presents an overview of the proposed service discovery protocol based on a grid architecture called SGrid. Every node adopts the periodic “hello” mechanism to announce its location, which is known via the Global Positioning System (GPS) or other location-aware devices. Every node acquires one-hop neighbor information following a “hello” period. The position information is then transformed into a 2-D logical grid. One node is elected in each grid cell to cache the shared service descriptions.

This paper proposes a hierarchical grid structure to register and discover services, as shown in Fig. 2. A provider that wishes to share its service to the MANET registers the service along a register trajectory. The directories in the register trajectory need to cache the service description. A requestor that wishes to access a service first checks whether it has the service information. If it does, the node requests the service to the provider. Otherwise, it queries the directory in its cell about the service information. If there is no any information in the directory, then it discovers the service along the discovery trajectory. The requestor obtains the reply as the discovery trajectory meets the register trajectory. Fig. 2 reveals that the node q obtains the service description from node r. The service description includes the information from provider p.

In this protocol, the register and discovery packets are not flooded to the entire network. Instead, the service is registered to the highest grid level along a predefined trajectory. The requestor only discovers the service information toward the maximum grid level. The predefined transmission trajectory can reduce unnecessary communication cost. Additionally, the requestor can quickly discover the service and efficiently decrease the discovery cost, because the maximum grid level is at the center of the network. The service is registered along the grid level, so this special registration route can make the node that is close to the provider quickly find the service. Moreover, the service description is only cached in the directories, thus lowering the storage cost.
Assume that every node has an omnidirectional antenna and the same transmission power. The link between two nodes is bidirectional, i.e., node A can hear node B, and node B can also hear node A. All nodes share the same communication channel (e.g., same frequency band, spreading code, and frequency hopping pattern) to send and receive packets. Hence, no node can send and receive at the same time. Every node has a unique ID, e.g., an Internet protocol or a medium access control (MAC) address, and is equipped with a device, e.g., a GPS receiver or other location-aware devices. To express the service types, a common ontology is known by all nodes. The grid model has been presented in [17]–[19]. The geographical area of the MANET is partitioned into a 2-D logical grid, as illustrated in Fig. 3. The grid has \( X \times Y \) cells, and every grid cell is a square of size \( d \times d \). Suppose that \( \text{grid}(x,y) \) denotes the grid-coordinates of cell. This protocol is implemented in a fixed area, including colloquia, zoos, botanical gardens, disaster areas, exhibitions, and metropolitan areas so that the origin of coordinates and the scale of network are regular parameters. Every node obtains these parameters when it joins a MANET.

Therefore, the physical coordinates can be transferred to the grid coordinates via these parameters.

In each grid cell, a node is elected as a directory. The cell size affects the number of directories. Let \( R \) indicate the transmission range and \( d \) indicate the side length of the cell. To transform from physical coordinates to grid coordinates, the relationship between \( d \) and \( R \) is defined as in Fig. 4. Simulation results reveal that \( d = R/\sqrt{2} \) yields the fewest directories. If \( d < R/\sqrt{2} \), then the number of cells increases. The number of directories also increases as the number of cells increases. In this protocol, every node adopts the periodical “Hello” mechanism to collect one-hop neighbor information. The nodes of a cell use this information to elect the directory. A cell might elect more than one directory if \( d > R/\sqrt{2} \). Fig. 4(a) demonstrates that the communication range \( R \) cannot cover the entire cell when the node is on the edge of the cell. Fig. 4(b) reveals that the communication range of a node can cover the entire cell at any place when \( d = R/\sqrt{2} \).

\[ d = R/\sqrt{2} \]

C. Notations

This section defines some notations that were adopted throughout this paper. The scenario is that the requestor wants to access a special service that is provided by a provider, \( N \). The ratio between the requestor and the provider is \( d/R \). The result reveals that the communication range of a node can cover the entire cell when the node is on the edge of the cell. Fig. 4(b) demonstrates that the communication range \( R \) cannot cover the entire cell when the node is not on the edge of the cell. Fig. 4(b) reveals that the communication range of a node can cover the entire cell at any place when \( d = R/\sqrt{2} \).

\[ d = R/\sqrt{2} \]

D. Hierarchical Grid Architecture

This section presents the hierarchical grid architecture for efficiently registering and discovering services. Two terms, i.e., grid level \( L \) and zone size \( (Z_x \times Z_y) \), are applied to form a hierarchical grid, as shown in Fig. 5(a), where \( L, Z_x, \) and \( Z_y \) are 3, 1, and 1, respectively. The grid level is from 0 to \( L \). The area of grid level 0 is also called zone, and the zone is composed of \( Z_x \times Z_y \) cells. The grid consists of \( X \times Y \) cells, where \( X = (1 + Z_x) \times 2^L - 1 \) and \( Y = (1 + Z_y) \times 2^L - 1 \). Fig. 5(b) illustrates an example for different zone sizes, where \( Z_y \) is varied from 1 to 3. This hierarchical grid architecture is easy to adapt to any scale of network. Hence, SGrid is appropriate for a predefined public place.

Assume that every node obtains the network information that includes the physical coordinates of \( \text{grid}(0,0) \).
A designated grid level input of Algorithm 2 comprises the grid coordinates of cell and node \( n \), level \( L_2 \) to derive the region of level 2. Algorithm 2 (see Fig. 8) to identify the register direction for service. This paper presents \( \text{Algorithm 1 (see Fig. 6), where these parameters. The grid level of a cell is determined using it can transform the coordinates to grid coordinates by using}

When a node acquires it or its neighbor’s physical coordinates, the grid level of a cell is determined using Fig. 7. Region of level 4. The region is located at northeast of level 4. region \( D \). That is, the grid \( (4,3) \) is located at region \( D \) of grid level 2.

**IV. Service Discovery Protocol**

This section presents the details of SGrid, including the directory selection, service register, and service discovery phase. This paper designs a service discovery protocol for efficiently raising the discovery success ratio and lowering the resource consumption. Furthermore, the proposed protocol is scalable and can be applied to any fixed public place.

Each node maintains three tables: 1) \( \text{neighbor} \); 2) \( \text{service} \); and 3) \( \text{route} \). The \( \text{neighbor} \) table records the neighbor information, including the \( xy \)-coordinates and the directory state of neighbors. The \( \text{service} \) table records the available service descriptions. The routing information is recorded in the \( \text{route} \) table. Every node communicates with other nodes using the routing information. If a node does not have a route to the destination node but knows the location of the destination node, then it communicates with the destination node by the geographical routing protocol.

**A. Directory Selection Phase**

To update the node location in the mobility environment, every node periodically broadcasts a Hello message to its neighbors. Every node also applies this message to obtain the neighbors’ state and the cell information of its neighbors. The format of a Hello message is defined as [Hello(\( id, x_{id}, y_{id}, state_{id} \)], where \( id \) denotes the unique ID of the broadcasting node, \( x_{id}, y_{id} \) denote the \( xy \)-coordinates, and \( state_{id} \) is either a Directory \( D \) or a Normal \( N \) node. A node that receives a hello message from its neighbor records the “hello” information into its neighbor table.

The directory is responsible for caching the available service descriptions, but it is expensive when it acts as a directory by a regular node in the mobility environment, because the directory has to constantly update its location for all nodes as it moves. A dynamic directory maintenance scheme is presented to overcome this problem. A node in each grid cell is elected as a directory if it has sufficient resources. Every node obtains the current directory information through the “hello” message. If \( d < R/\sqrt{2} \), then a node can perceive whether a directory exists in a cell, thus ensuring that only one directory exists in a cell. The node that is close to the center of the cell is elected as the directory, whereas no directory exists in the cell. Fig. 9 presents an example for directory selection, where we know the region of level 2 for \( i \). If the input values are \( C_i, x = 4 \), \( C_i, y = 3 \), and \( lv = 2 \), then the output is \( \text{region} D \). That is, the grid \( (4,3) \) is located at region \( D \) of grid level 2.
assume that every node has broadcast a hello message, and no directory exists in the cell [Fig. 9(a)]. For instance, node $A$ broadcasts $Hello(A, x_A, y_A, N)$ to announce the node information. Every node can learn from the “hello” message whether a directory exists in this cell. When node $A$ perceives that no directory exists in this cell, and it is closest to the center of the cell, it acts as the directory and informs its neighbors about this information at the next “hello” period by broadcasting $Hello(A, x_A, y_A, D)$. However, mobility is a significant issue in MANETs, because the grid is a virtual framework that cannot restrict node movement within a cell. Hence, a node may enter or leave a cell at any time. To decrease the change frequency of a directory, the directory is reelected when the original directory has already moved out the cell or if it does not have adequate resources to serve as a directory. When a directory $n$ leaves the original cell, another node is reelected as the directory to supersede $n$ in the original cell. In Fig. 9(b), although node $A$ is not the node that is closest to the center of cell, it is still in the cell; therefore, node $A$ is still the directory. Node $B$ knows that node $A$ is still in the cell by hello, so it still broadcasts $Hello(B, x_B, y_B, N)$ to its neighbors. If node $A$ moves out of the original cell, then it becomes a normal node [Fig. 9(c)], and node $B$ becomes the new directory. This approach also supports another similar scenario, i.e., directory failure.

### B. Service Register Phase

This section presents the provider how a service can be registered. Broadcast methods for registering services are very costly, so this paper addresses a geographical transmission scheme, i.e., a provider registers its service along a register trajectory. The service description is registered to the center line of the network after performing this phase. The directories of cells need to cache the service description into their service tables during the register process. Two basic transmission directions are north–south and east–west. The register trajectory is formed in accordance with the region and the level of the provider’s cell.

The rules for forming a register trajectory are presented as follows. The grid level $lv$ of the provider determines the starting transmission direction. If $lv = 0$, then the initial transmitting direction conforms to the region of $lv$, in which the provider is located. If $lv > 0$, then the registered packet is sent along the grid level $lv$. The transmission direction changes when a packet is sent from grid level $l$ to $l + 1$. Fig. 10 indicates four register trajectories for grid level 0, where $Z_x = Z_y = 3$, and $L = 1$. The starting transmission direction is east–west if the cell is in the region $A/C$ of level 1; otherwise, it is north–south.

A provider in $grid(2,0)$, as depicted in Fig. 10(a), first chooses two geographical points as the destinations. It then adopts geographical routing to deliver the register packet there. The two geographical points are at the west of $grid(1,6)$ and at the east of $grid(4,6)$. In geographical routing, a forwarding node chooses a relay node, which is the node closest to the destination within the transmission range of the forwarding node, to deliver the packet to the geographical point. The relay node selection method will later be described in detail. The register packet is broadcast along the register trajectory. When a node in the cell of geographical point receives the packet, and the level of cell is higher than that of the forwarding node’s cell, it changes the original transmission direction and specifies two new geographical points. For instance, the transmission direction in $grid(4,6)$ is changed from east–west to north–south, and the two new geographical points are at the south of $grid(4,1)$ and the north of $grid(4,7)$, respectively.

If the provider is at the cell of grid level 1 (or higher level), then the register trajectory is along the direction of the grid level. Assume that a provider is at $grid(4,5)$, as shown in Fig. 10(a). $grid(4,5)$ is in level 1; therefore, the starting transmission direction is north–south along the grid level. In a special situation, the provider $p$ is at the intersection of grid level (i.e., $grid_p.xlv = grid_p.ylv$). The provider has the choice of two registering directions. The registering direction at the intersection of the grid level is defined as north–south.

The selection of the relay node is discussed in this section. In traditional geographical routing, the node that is closest to the geographical point is specified as the relay node. However, this approach may make the transmission deviate from the expected register trajectory. Hence, choosing the node that is closest to the register trajectory as the relay node is considered to keep the transmission along the expected register trajectory. The transmission may be inefficient as the relay and the forwarding nodes are at the same cell. To enhance the transmission efficiency and the deviation problem of transmission, the conditions under which a node is close to the register trajectory and to the geographical point are simultaneously considered in the selection algorithm. The priority of the node that is nearest the transmission trajectory is higher than that of the node nearest the geographical point.

Fig. 11 shows the relay node selection algorithm, where $n$ denotes the forwarding node, $dir$ denotes the transmitting direction, $gp$ denotes the geographical point, $C_{gp}$ denotes the cell of $gp$, $rel_{dir}$ denotes the selecting relay node toward the $dir$, and $SelR((C_{a,x}, C_{a,y}), (C_{b,x}, C_{b,y}), gp)$ is a function for choosing the relay node that is closest to $gp$ from $cell_a$ to $cell_b$. If the input value of $SelR$ is $((C_{a,x}, all), (C_{b,x}, all), gp)$, then the selecting area forms a sector, as in Fig. 12(b). If the input...
Algorithm 3: SelRel(Ca,x, Ca,y, gp, dir)

1 rel0 = Null;
2 // selecting relay node from SR-A //
3 if (dir = east) then rel1 = SelR(Ca,x, Ca,y), (Ca,x+1, Ca,y), gp)
4 if (dir = west) then rel1 = SelR(Ca,x, Ca,y), (Ca,x-1, Ca,y), gp)
5 if (dir = north) then rel1 = SelR(Ca,x, Ca,y), (Ca,x, Ca,y+1), (Ca,x, Ca,y-1), gp)
6 if (dir = south) then rel1 = SelR(Ca,x, Ca,y), (Ca,x, Ca,y+1), gp)
7 // selecting relay node from SR-B //
8 if rel = Null then
9 if (dir = east) then rel = SelR(Ca,x, Ca,y), (Ca,x+1, Ca,y), gp)
10 if (dir = west) then rel = SelR(Ca,x, Ca,y), (Ca,x-1, Ca,y), gp)
11 if (dir = south) then rel = SelR(all, Ca,y), (all, Ca,y+1), gp)
12 if (dir = north) then rel = SelR(all, Ca,y), (all, Ca,y-1), gp)
13 // selecting relay node from SR-C //
14 if rel = Null then
15 if (dir = east) or (dir = west) then rel = SelR(Ca,x, Ca,y), null, gp)
16 if (dir = south) or (dir = north) then rel = SelR(all, Ca,y), null, gp)
17 // selecting relay node from SR-D //
18 if rel = Null then
19 if (dir = east) or (dir = west) then rel = SelR(Ca,x, Ca,y), null, gp)
20 if (dir = south) or (dir = north) then rel = SelR(all, Ca,y), null, gp)

Algorithm 4: Register Process

1 rec = false;
2 node i receives a packet reg from node j;
3 i caches j as next node to pvd in RT;
4 if i is a directory then i caches locpvd and sdipvd in ST;
5 if i is rel and rec = false then
6 rec = true;
7 if i is in the cell of gp then
8 if changed(dir) = false then i broadcasts reg;
9 else i computes gp_and gp_dire;
10 SelRel(Ca,x, Ca,y, gp, gp_and, dire);
11 SelRel(Ca,x, Ca,y, gp, gp_dire);
12 i delivers reg to ngr and mgr;
13 else
14 SelRel(Ca,x, Ca,y, gp, d);
15 i continues to deliver reg to gp;
16 // checking the transmitting direction is needed to change //
17 bool changed(dir) {
18 // computing td_i //
19 if dir = east or dir = west then td_i < 0;
20 else td_i < 1;
21 // computing td_dire //
22 if gp_and = 0 then
23 if C_i.region = A or C_i.region = C then td_dire < 0;
24 else td_dire < 1;
25 else
26 if C_i.x < 2C_i.y tol ove tx then td_dire < 1;
27 else td_dire < 0;
28 // comparing with td_i //
29 if ((C_i.x < 2C_i.y ) and (td_dire < td_i)) or
30 ((C_i.x < 2C_i.y ) and (td_dire = td_i)) then
31 return false;
32 else return true;
33 }
Fig. 15. Example for register process.

\( pvd, \text{loc}_{pvd} \) denotes the location of \( pvd \), \( fwd \) denotes the forwarding node, \( \text{loc}_{fwd} \) denotes the location of \( fwd \), \( rel_x \) denotes the relay node on the east trajectory, and \( gp_x \) denotes the geographical point on the east trajectory (i.e., \( w \) is the west trajectory, \( s \) is the south trajectory, and \( n \) is the north trajectory).

When node \( i \) receives a register packet \( reg \) from \( j \), it sets \( j \) as the next node to \( pvd \), and this routing information is cached in routing table \( RT \). Node \( i \) utilizes this routing to transmit packets when it wants to communicate with \( pvd \). If node \( i \) is a directory, then it stores the location information of \( pvd \) and the service descriptions in its service table \( ST \). If node \( i \) is one of the relay nodes \( \text{rel}_d \) of \( reg \), and it receives this packet the first time, then it checks if it is in the cell of geographical point \( gp_d \), where node \( \text{rel}_d \) is the relay node on the \( d \) trajectory, and \( gp_d \) is the geographical point on the \( d \) trajectory. If node \( i \) is not in the cell of \( gp_d \), then it continues to deliver \( reg \) to the geographical point. Otherwise, node \( i \) calls the \( \text{changeTd} \) function to check whether the transmission direction should be changed. If \( \text{changeTd}(d) = \text{false} \), then node \( i \) broadcasts the register packet \( reg \) to its neighbors. Otherwise, node \( i \) calculates two new geographical points, i.e., \( gp_{d1} \) and \( gp_{d2} \), and then, it delivers the registered packet toward the new geographical points.

Fig. 15 shows an example for the service registration process, where a provider \( P \) is in \( \text{grid}(2,6) \), \( Z_x = Z_y = 3 \), and \( L = 1 \). \( \text{grid}(2,6) \) is in region \( A \) of level 1, so the transmission direction of \( P \) is east–west. The geographical points \( gp_w \) and \( gp_p \) are at the west of \( \text{grid}(1,6) \) and the east of \( \text{grid}(4,6) \), respectively. Provider \( P \) issues \( \text{reg}(\text{register}, P, \text{sd}_p, \text{loc}_p, P, \text{loc}_p, F_2, \text{gp}_p, F_1, \text{gp}_w, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}) \) to register the service. The transmission direction is east–west, so the fields \( \text{rel}_x \), \( \text{gp}_x \), \( \text{rel}_n \), and \( \text{gp}_n \) are \( \text{NULL} \). Nodes \( F_2 \) and \( F_1 \) are chosen as the relay nodes of \( gp_p \) and \( gp_w \), respectively. Node \( F_2 \) continues to forward the packet to \( F_3 \). \( F_3 \) is in the cell of \( gp_w \), so it broadcasts \( \text{reg}(\text{register}, P, \text{sd}_p, \text{loc}_p, F_1, \text{loc}_f_1, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}) \) to its neighbors and does not assign any relay node. Node \( F_3 \) obtains the value \textit{true} from the \textit{changeTd} function when it receives \( reg \) from \( F_2 \). Therefore, \( F_3 \) has to change the transmission direction and recompute two new geographical points, which are at the south of \( \text{grid}(4,1) \) and the north of \( \text{grid}(4,7) \). Node \( F_3 \) issues \( \text{reg}(\text{register}, P, \text{sd}_p, \text{loc}_p, F_3, \text{loc}_f_3, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}, \text{NULL}) \). This process is repeated until \( F_4 \) and \( F_9 \) receive this packet.

The maintenance of SGrid adopts the soft mechanism. The service provider periodically registers the service along the register trajectory. The directory caches the register within a period \( T_r \). The register becomes invalid after the expiry period if it is not updated. If a directory leaves the original cell, then it still maintains the service descriptions in its service table until the period \( T_r \) has expired. If many providers share the same service type, then the network consumes more resources to register the same service. To solve this problem, a relay node stops forwarding the register packet if it has the same service description. Three time thresholds, i.e., refresh time \( T_r \), periodical register time \( T_p \), and expiration time \( T_e \), are defined, where \( T_e < T_p < T_r \). A provider registers a service every \( T_p \) s. The service description is cached in the directory’s service table within \( T_e \) s. If a relay node receives a register packet from another provider, and it has the same service description, then it verifies whether the same service type has been registered to the same trajectory within \( T_e \) s. If yes, then it is unnecessary to again deliver the packet for the relay node.

Fig. 16 depicts an example for a service description update. A provider \( P_1 \) registers a service along trajectory at \( T_1 \) (see Fig. 16(a)). Assume that another provider \( P_2 \) registers the same service after \( P_1 \). If the directory \( D \) in \( \text{grid}(2,5) \) receives the register packet from \( P_2 \) at \( T_2 \), then it does not need to deliver the packet to the same register trajectory [see Fig. 16(b)], because the previous register does not exceed \( T_1 \). If \( T_2 > T_1 + T_r \), then \( D \) has to refresh the service register along the trajectory [see Fig. 16(c)].

C. Service Discovery Phase

The service discovery process is executed when a requestor cannot search out the available service information, including the ID and location of the provider, and the service type in its service table. Due to two characteristic of this protocol, the requestor can quickly discover the service and effectively decrease the discovery cost. The first character is that the network is located within a predefined rectangular area, so every node knows the network range. The second character is that the service description is registered to the center line of the network along the register trajectory. That is, the requestor can quickly discover the service and effectively decrease the discovery cost. The first character is that the network is located within a predefined rectangular area, so every node knows the network range. The second character is that the service description is registered to the center line of the network along the register trajectory. That is, the requestor can quickly discover the service and effectively decrease the discovery cost.

Fig. 17 illustrates the discovery trajectories, where grid level 1 is the maximum level. In Fig. 17(a), a requestor \( R_A \) is at the region \( A \) of the level 1, and therefore, it discovers the
service toward the east and the south. The geographical point computation and the relay node selection for discovery process are the same as those of the registration process. The discovery process is stopped at the maximum grid level; therefore, the discovery range is confined to a quarter of the network. If a directory receives the discovery packet, and it has the available service description, then it responds with a reply packet that contains the service description to the requestor.

The format of discovery packet is defined as
\[ \text{discovery(type, req, sd_{req}, loc_{req}, fwd, loc_{fwd}, rel_e, gp_e, rel_w, gp_w, rel_s, gp_s, rel_n, gp_n)}, \]
where \( \text{type} \) denotes packet type, \( \text{req} \) denotes the requestor, \( \text{sd}_{\text{req}} \) denotes the desired service description, \( \text{loc}_{\text{req}} \) denotes the location of \( \text{req} \), \( \text{fwd} \) denotes the forwarding node, \( \text{loc}_{\text{fwd}} \) denotes the location of \( \text{fwd} \), \( \text{rel}_e \) denotes the relay node in the east trajectory, and \( \text{gp}_e \) denotes the east geographical point (\( w \) denotes the west trajectory, \( s \) denotes the south trajectory, and \( n \) denotes the north trajectory). The format of the discovery reply packet is
\[ \text{dis\_rep(type, req, sd_{req}, loc_{req}, fwd, loc_{fwd}, pwd, loc_{pwd}, rel)}, \]
where \( \text{pwd} \) denotes the provider, \( \text{loc}_{\text{pwd}} \) denotes the location of \( \text{pwd} \), and \( \text{rel} \) denotes a relay node. The format of service request packet is
\[ \text{request(type, req, sd_{req}, loc_{req}, fwd, loc_{fwd}, pwd, loc_{pwd}, rel)}. \]

Fig. 18 presents an example for service discovery, assuming that provider \( P \) has registered a service to network, and requestors \( Q \) and \( R \) want to access a service that \( P \) provided. Requestor \( R \) is a directory for \( P \), so it can immediately obtain the provider information and immediately request \( P \) to provide service. The routing between \( R \) and \( P \) can adopt any geographical routing protocol. Another requestor \( Q \) does not have the available service information, so \( Q \) issues a packet broadcast along the discovery trajectory to discover service. \( Q \) is located in region \( B \) of level 1, and the transmission directions of the packet are toward the west and the south. \( Q \) calculates two geographical points of discovery trajectory and delivers a discovery packet toward the area. The selected relay nodes of \( Q \) are \( F_2 \) and \( F_3 \). \( F_5 \) does not have the available service description; therefore, it continues to deliver packet toward the south. \( F_2 \) is a directory and has the service description; therefore, it stops forwarding the packet and returns a discovery reply packet to \( Q \). Aside from \( F_2 \), \( Q \) also receives the reply packets from \( F_1, F_3, F_4, \) and \( F_7 \). When a requestor obtains the service provider information, it requests the provider to provide service. The routing between requestor and provider can apply any existing geographical routing protocol. When a provider receives a \( \text{req} \) packet, it returns a \( \text{req\_ack} \) packet, which contains the service port and informs the requestor to allow access to a service to the requestor. The discovery process is finished when the requestor receives \( \text{req\_ack} \).

D. Enhanced Register Phase

Due to node mobility, some areas become empty areas, i.e., they do not contain nodes. If the register packet encounters an empty area, then the register trajectory is broken. This section presents an obverse register trajectory to avoid empty areas. The obverse register trajectory (see Fig. 19) is the opposite of the normal register trajectory (see Fig. 10). The obverse register trajectory means changing the transmission direction, e.g., changing east–west to north–south or changing north–south to east–west. The relay node turns the transmission direction to the obverse/normal register trajectory when the normal/obverse register trajectory is broken. Fig. 20 illustrates an example of an empty area, assuming that the provider is in \( \text{grid}_{(1,6)} \), and an empty area is located at \( \text{grid}_{(4,5)}, \text{grid}_{(4,6)}, \) and \( \text{grid}_{(4,7)} \).
When a relay node in grid\(_{(3,6)}\) cannot find the next relay node to the geographical point, it turns the register trajectory to the obverse direction. If the obverse register trajectory is not performed, then the requestor might miss the service. Two new geographical points are obtained in the north of grid\(_{(3,7)}\) and the south of grid\(_{(3,4)}\). The register packet is then sent along the obverse trajectory. A requester in grid\(_{(6,6)}\) obtains the service information from a directory in grid\(_{(6,4)}\) as in Fig. 20(b). This enhanced register process raises the discovery and solves the empty area problem in a network.

V. ANALYSIS AND SIMULATION RESULTS

This section describes the analysis and simulation conducted to measure the performance of SGrid and GCLP [21]. The number of registered directories, register cost, and discovery cost were evaluated in the analysis. In the simulation, the system performance was measured with different grid sizes, node mobility, node densities, hello intervals, mobility models, and geographical routing protocols. Five criteria were adopted to judge the performance and overhead for protocols: 1) the success ratio; 2) the directory ratio; 3) the control overhead; 4) the discovery time; and 5) the number of packet types. The success ratio is derived as the ratio of requestor that succeeded in accessing the service. A high success ratio demonstrates that the service discovery protocol can aid the requestor to access service. The directory ratio is the number of registered directories divided by all nodes. A low directory ratio reduces the storage cost for registers. In SGrid, every cell has only one directory. If a directory receives a register packet from a provider, then it needs to cache the register. When a register packet is transmitted, \(k\) neighbor directories hear this packet. Fig. 21(a) illustrates an example for \(k\) neighbor directories, where \(k = 2\). The register trajectory contains \(C\) cells in (1). The number of registered directories is equal to the number of cells that the register trajectory passes [Fig. 21(b)]. Equation (2) calculates the average overhead of a registered directory for SGrid \((D_{SGrid})\) as

\[
C \approx (X + Y)/2 \sum_{i=0}^{L} 1/2^i
\]

\[
D_{SGrid} = (k + 1) \times C, \quad \text{where } k = \lceil 2 \times R/d \rceil.
\]

In GCLP, the forwarding node broadcasts the register packet along the register trajectory. In the register trajectory, the node that is farthest from the forwarding node is specified as the relay node. The transmitting hop count is derived with (3). The \(Hop_x\) and \(Hop_y\) indicate the number of nodes that forward the register packet along the \(x\)- and \(y\)-coordinates, respectively. The nodes that received the register packet have to act as directories. Therefore, the nodes that are near the register trajectory are the registered directories, as displayed in Fig. 21(c). To compute the number of registered nodes, the area of all registering regions is calculated. The forwarding node overlaps the registering regions of other nodes. The RA denotes the registering area for each node and equals the entire registering area divided by the number of nodes. In Fig. 21(d), the RA is marked green and is determined by (4). The formula for the total registering area \((TRA)\) is calculated and listed in (5), and an example of \(TRA\) is shown in Fig. 21(e). The average number of nodes for each hop is \(N/(Hop_x \times Hop_y)\). The average

\[
|TRA| \approx X \times Y \times \frac{1}{N} \times \frac{1}{(Hop_x \times Hop_y)}
\]
overhead of the registered directory for GCLP ($D_{GCLP}$) is given by
\[
Hop_x = \lceil H/T \rceil, \quad Hop_y = \lceil W/T \rceil
\]
\[
RA = 4 \times \int_0^{T/2} \sqrt{R^2 - x^2} dx
\]
\[
d_x = 4 \times \left( \frac{T}{4} \sqrt{R^2 - (T/2)^2} + \frac{R^2/2}{2} \sin^{-1} \frac{T/2}{R} \right)
\]
\[
TRA \approx RA \times (Hop_x + Hop_y) - 4 \times (R^2 - (T/2)^2)
\]
\[
D_{GCLP} = TRA/T^2 \times N/(Hop_x \times Hop_y).
\]

The average and worst discovery costs for SGrid and GCLP are given by (9)–(12), respectively. We have
\[
C_{reg_{SGrid}} = (Hop_x + Hop_y)/2 \times \sum_{i=0}^{L} 1/2^i
\]
\[
C_{reg_{GCLP}} = Hop_x + Hop_y.
\]

The discovery packet overhead is utilized to measure the service discovery cost of the requestor. The service discovery cost includes the discovery packet and dis_rep packet. In SGrid, the service description is registered to the center line of the network, and the requestor has known the location of L so that the requestor can directly discover services toward the maximum level L. The discovery range is restricted to the region of level L, so the discovery cost is significantly lowered. The average and worst discovery costs for SGrid and GCLP are given by (9)–(12), respectively. We have
\[
C_{dis_{SGrid_{Avg}}} = (Hop_x + Hop_y)/4
\]
\[
C_{dis_{SGrid_{Worst}}} = (Hop_x + Hop_y)/2
\]
\[
C_{dis_{GCLP_{Avg}}} = (Hop_x + Hop_y)/2
\]
\[
C_{dis_{GCLP_{Worst}}} = Hop_x + Hop_y.
\]

Fig. 22 shows the results of analysis for two cases. The network is 1200 m x 1200 m ($H \times W$) in case A and is 600 m x 1200 m in case B. Other parameters are given as follows: $N = 300$, $R = 250$ m, $L = 1$ to 3, $d = 88$ to 250 m, and $T = 180$ to 250 m. Fig. 22(a) and (b) reveals the directory ratio results for GCLP and SGrid, respectively. The directory ratio denotes the number of registered directories divided by the number of nodes. The node density of the small network (i.e., case B) is higher than that of the large network (i.e., case A). Hence, the directory ratio of the small network is greater than that of the large network. The average of the maximum transmission distance between two nodes $T$ slightly influences the directory ratio. The directory ratio of GCLP at $T = 200$ is compared with that of SGrid. The comparison results show that the directory ratio of SGrid is less than that of GCLP, and therefore, SGrid effectively reduces the register storage cost. Furthermore, the directory ratio of SGrid for a different case is better than that of GCLP. The directory ratio of the small network is lower than that of the large network, because the small network had fewer cells in the register trajectory, and each cell had only one directory to record the service description. The directory ratio of SGrid declined when the cell size $d$ rose. Fig. 22(c) and (d) presents the results of the register cost and discovery cost, respectively. The cell size does not affect the register cost, because the main factors that influence the register cost are node deployment and level quantity. The node deployment affects the transmitting hop count, and the level quantity affects the length of the transmission trajectory. The register cost decreases when $T$ increases. The impact of level is opposite to that of $T$. The discovery cost is also influenced by $T$. 

![Figure 22](image-url)
B. Simulation Model

The Carnegie Mellon University (CMU) wireless extensions [24] of the ns2 [25] simulator were adopted to estimate the performance of protocols. The CMU extensions provide IEEE 802.11 radio and MAC model simulation [26] for ns2 nodes. The radio range is approximately a disc, and the radius is set to 250 m. The communication bandwidth is 2 Mb/s. Five runs are simulated, with each using the same parameters. Every run is simulated for 300 s. The nodes are deployed at random in a square area of 1200 × 1200 m. The number of nodes is varied from 150 to 300. The grid level L is set to 1. There exist ten service types in the network, and three nodes provide the same service type. Any node may be a requestor for a service type, except for the provider. The inquiring service and issuing time of requestors are generated at random. A requestor does not inquire the same service twice in a simulation run. The hello interval is set to 2 s, the allowed_hello_loss is set to 2, the register time \( T_r \) is set to 10 s, and the expiration time \( T_e \) is set to 20 s.

The protocols of SGrid and GCLP are simulated based on two mobility models: 1) the random waypoint (RWP) model [27] and 2) the Manhattan mobility model [28]. In RWP, one node specifies a random destination and moves at a constant speed. The speed is uniformly chosen between minimum and maximum speed. After a node reaches the destination, it waits for a pause time before another random destination is specified. The pause time is varied from 0 to 30 s. The Manhattan mobility model simulates the mobile behavior of nodes for urban areas. The movement of one node is restricted to horizontal and vertical streets. A mobile node can turn left, turn right, or go straight at a certain intersection. The probabilities of going straight, turning left, and turning right are set to 0.5, 0.25, and 0.25, respectively.

Service discovery protocols with different geographical routing are also simulated. The routing protocol is employed to transmit a service request from a requestor to a service provider, whereas the requestor obtains the service provider information. Location-aided routing (LAR) is first presented by Ko and Vaidya [29]. This simulation adopts scheme 2 of LAR. If a packet is sent from \( F \) to \( D \), and the distance from \( N \) to \( D \) is shorter than that from \( F \) to \( D \), then node \( N \) forwards it. That is, the packet is broadcast in a small “request zone.” Another routing protocol, which is called shortest distance routing (SDR), forwards a packet to the destination along the shortest distance. This scheme has a lower transmission cost than LAR, but its routing is easier to break. Retransmission is permitted if the requestor has not received the reply from the provider. The retransmission interval is set to 5 s, and the allowed_routing_error is set to three times in the simulation. LAR does not support retransmission method, since it transmits packets using local flooding.

C. Evaluating Different Cell Size

The performance is first evaluated with different cell sizes, as illustrated in Fig. 4. Nodes in the same cell are guaranteed to communicate with each other with a transmission range of 250 m and a cell size of 176 m. When the grid level is fixed, the cell size influences the zone size, as shown in Table I, where \( L = 1 \). The highest grid level is located at the center of the network as many as possible. Fig. 23 presents the performance for different cell sizes, where 300 nodes are deployed in the network. The routing protocol SDR is applied, the mobility model is RWP, the maximum velocity of node is varied from 5 to 30 m/s, and the minimum node velocity is set to 0 m/s. SGrid-d represents that SGrid is implemented with cell size \( d \).

Fig. 23(a) shows the success ratio at different cell size values. The success ratio dropped as the maximum velocity increased. The network topology is quickly changed with high node mobility so that the success ratio is dropped. Fig. 23(b) presents the directory ratio, which shows the percentage of nodes that have to cache service descriptions. The simulation results are similar to those of the analysis in the directory ratio, except in the case of SGrid-250. The directory ratio rose as the grid size decreased. The directory ratio is lowest when the cell size is equal to 176 m. The directory ratio of SGrid-250 is higher than those of SGrid-176 and SGrid-158 in the simulation results, because the node cannot obtain the information of all nodes in the same cell by one-hop hello messages. One cell may contain more than one directory when \( d = 250 \) m. The maximum cell size for communication with other nodes in the same cell is found to be 176 m.

Fig. 23(c) shows the total control overhead, including all control packets. The total control overhead of SGrid-176 and SGrid-158 did not increase as the node velocity rose. Other SGrid-ds mainly increase the overhead for discovery. The discovery cost in Fig. 23(d) is similar to that in Fig. 23(c) in the curve. The discovery cost increased as the node velocity rose because of the increased number of reply packets for discovery (dis_rep). If the cell size is small, then the service is registered in more directories than when the cell size is large. A node with a higher velocity more frequently moves to other cells, and thus, the directory with a higher velocity is also frequently reselected. Node \( n \) retains its service descriptions until the expiration time \( T_e \), expired when it is relieved of a directory. The expiration time \( T_e \) is larger than the register time \( T_r \), so node \( n \) may maintain an incorrect service description if the time that it keeps a register is longer than \( T_r \). If a requestor obtains the service information from \( n \), then it may not successfully access the service. In this case, the requestor has to rediscover the service. SGrid with a small cell size has a higher discovery cost than one with large cell size, as revealed in Fig. 23(e). Fig. 23(f) presents the discovery times, validating this explanation. SGrid with a small cell size takes more time to successfully access services than that with a large cell size.

This paper has investigated the performance of SGrid for different cell size. The SGrid-176 discovers service faster and saves more register storage than the other SGrids. It also lowers the discovery cost as the node speed increased. Therefore, SGrid-176 is selected for comparison with GCLP.
D. Comparing With GCLP

This section shows the comparison of SGrid with GCLP [21]. SGrid is compared with GCLP, because GCLP is a geography-based service discovery protocol, in which the service is registered along a predefined trajectory, and the node is equipped with the location-aware devices. The performance of protocols is measured with different scenarios in terms of node density, routing protocol, and mobility model.

1) Varying Minimum Node Speed: The performance of the protocols is measured at different minimum node speeds. The number of nodes is 150, the mobility model is RWP, and the routing protocol is SDR. The maximum node speed is 30 m/s, and the minimum node speed is varied from 5 to 30 m/s. The success ratio is dropped as the minimum node speed increased [see Fig. 24(a)]. The node mobility affects the routing stability, which sometimes causes service discovery to fail in a high mobility environment. The success ratio of SGrid is higher than that of GCLP, particularly in high mobility. The success ratio of SGrid at 30 m/s is 5.89% higher than that of GCLP. The discovery ratio of SGrid is lower than that of GCLP [see Fig. 24(b)], and these results are the same as those in the analysis. In GCLP, many nodes have to cache the register. In SGrid, only one directory has to cache the register for each cell. Therefore, SGrid saves more register storage than GCLP. Fig. 24(c) and (d) presents the total control and packet overheads, respectively. GCLP has a high discovery cost, including many dis and dis_rep packets, because the service discovery is performed along four trajectories, and many directories are needed to reply with the service information to the requestor after receiving the dis packets. SGrid has a much lower discovery cost, because the service discovery is restricted to a region of the highest grid level, and only a few dis_rep packets are generated. The discovery cost of GCLP rose as the node speed increased, because the rediscovery process increased. Although the requestor has the service information, it has to discover the service when the information is out of date. If the service information is within $T_e$ but is not refreshed by the next register, the information is out of date. In this case, the requestor cannot find the provider or successfully access the service. In SDR routing, the requestor can rediscover service three times if it does not access the service within 5 s. The discovery time of GCLP rose as the node speed increased, as observed in Fig. 24(e), particularly in high mobility.

2) Varying Maximum Node Speed: The performance of the protocols is measured at different maximum node speeds. The number of nodes is set to 300, the mobility model is RWP, and the routing protocol is SDR. The minimum node speed is 0 m/s, and the maximum node speed is varied from 5 to 30 m/s. The results of all criteria in Fig. 25 are the same as those in Fig. 24.

3) Varying the Number of Nodes: The performance is studied at different numbers of nodes. The number of nodes is varied from 150 to 300, the mobility model is RWP, and the routing protocol is SDR. The minimum node speed is 0 m/s, and the maximum node speed is varied from 5 to 30 m/s. The results of all criteria in Fig. 25 are the same as those in Fig. 24.

A high node density led to many collisions, thus affecting the success ratio. The directory ratio declined as the number of
nodes increased, particularly in SGrid, as revealed in Fig. 26(b). Thus, SGrid is found to have a significantly register storage than GCLP. The total control cost increased as the number of nodes increased [see Fig. 26(c)]. The discovery cost is the main cost that influences the total control overhead. SGrid requires fewer discovery reply packets than GCLP, because the service description is registered to the head of cells in SGrid, which means that few nodes have to reply with service information to requestor. SGrid is found to access a service faster than GCLP [see Fig. 26(d)]. Although 66% of nodes have the service description, the descriptions of some nodes are out of date. A
node that detects incorrect information has to rediscover the service. Therefore, the discovery time of GCLP is longer than that of SGrid.

4) Simulation Results With the Manhattan Mobility Model:
The performance is studied with the Manhattan mobility model. The minimum node speed is 0 m/s, the maximum node speed is 30 m/s, the number of node is varied from 150 to 300, and the routing protocol is SDR. To simulate the behavior waiting traffic lights in the Manhattan mobility model, every node has a brief pause at an intersection of a road. The pause time is varied from 0 to 30 s and the same as RWP. A virtual grid street map is structured, including 16 streets from the east to the west and nine streets from the north to the south. The movement of nodes is restricted to the streets.

Every node movement distance between two pauses in the Manhattan mobility model is shorter than that in RWP, because each node has a pause at each intersection in the Manhattan mobility model, where the block size is set to 150 × 75 m. Thus, Manhattan has a steadier network topology than RWP. Hence, the success ratio with the Manhattan mobility model [see Fig. 27(a)] is higher than that with RWP [see Fig. 26(a)]. The simulation results of the Manhattan mobility model [see Fig. 27(b)] are similar to those of RWP [see Fig. 26(b)] in directory ratio. Fig. 27(c) presents the control overhead, and Fig. 27(d) presents the discovery time. SGrid is found to more efficiently perform discovery and save more discovery cost than GCLP.

5) Simulation Results With the LAR Protocol:
The performance is studied with the LAR protocol [29]. The routing
protocol is adopted to send a service request from a requestor to a service provider after a requestor obtains the service provider information. LAR, which is a geographical routing protocol, utilizes a local flooding zone to send a packet from a source to a destination. The minimum node speed is 0 m/s, the maximum node speed is 30 m/s, the number of node is varied from 150 to 300, and the mobility model is RWP. The pause time is varied from 0 to 30 s and the same as RWP. Rediscovery is not supported for LAR in the simulation, because LAR adopts a flooding scheme to send packets, which causes too much communication cost.

Fig. 28(c) shows that the main control packet cost is the request cost. The control overhead with LAR [see Fig. 28(c)] is higher than that with SDR [see Fig. 27(c)], particularly in request cost, which means that the different routing protocols affect the number of request packets. The success ratio of LAR [see Fig. 28(a)] is lower than that of SDR [see Fig. 27(a)], because LAR generated many packets to obstruct the network. Fig. 28(a) shows that the success ratio of SGrid is still higher than that of GCLP. The simulation results of LAR are similar to that of SDR in the directory ratio [see Fig. 28(b)]. A requestor
needs to wait for some time to receive the reply from the provider if the traffic of network is obstructed [see Fig. 28(d)]. The discovery time of SGrid at 300 nodes is slightly higher than that of GCLP. The collision declined the success ratio and obviously increased the discovery time. Additionally, SGrid has a higher success ratio than GCLP, which causes the discovery time of SGrid to increase.

6) Varying Hello Message Intervals: The performance is studied at different hello message intervals. The hello interval influences the update frequency and the accuracy of information, including the neighbor nodes and routing. Thus, these factors also affect the success ratio. The number of node is 150, the mobility model is RWP, the minimum node speed is 5 m/s, the maximum node speed is 30 m/s, and the routing protocol is SDR.

Fig. 29(a) shows that the success ratio decreased as the interval increased, i.e., interval = 1 s, because the long interval affected the accuracy of routing. The routing data may be inaccurate due to the lack of node update. However, frequent updates generated many “hello” messages to obstruct the network [see Fig. 29(d)]. The success ratio at interval = 2 s is better than at other intervals. SGrid slightly rose the directory ratio at interval = 5 s [see Fig. 29(b)], because infrequent node updates influenced the accuracy of directory selection. Thus, some cells have more than one directory. Fig. 29(c) shows that the control overhead of SGrid is lower than GCLP. High intervals increase the discovery cost [see Fig. 29(d)] because of the incorrect routing. The hello interval also influenced the discovery time [see Fig. 29(e)]. The shortest discovery time of SGrid and GCLP are both obtained at interval = 3 s. The rediscovery frequency and discovery time are increased if interval > 3 s. The collision and discovery time are increased if interval < 3 s.

SGrid, which adopts a varying trajectory to register and discover service, is different from GCLP, which adopts a criss-cross trajectory. Analytical and simulation results demonstrate that SGrid has better success ratio, directory ratio, control overhead, and discovery time than GCLP. SGrid can assist more requestors to obtain service than GCLP, thus more effectively discovering service. SGrid needs less register storage than GCLP, because only one directory has to cache the service description in each cell. In contrast, GCLP has to register the service to many directories near the four register trajectories. In additional, the discovery area of SGrid (i.e., one quarter of the network) is smaller than GCLP (i.e., an entire network). SGrid has a much lower total control overhead, in particular discovery cost, than GCLP; therefore, fewer packets reduce energy consumption. The requestor can quickly discover and access the service using SGrid, because the directory returns the correct service information to the requestor. Using fewer directories to manage the service description raises the accuracy of information. Thus, SGrid can provide stable fast efficient service discovery for users.

VI. CONCLUSION

This paper has proposed a service discovery protocol for MANETs called SGrid. The proposed protocol raises the discovery ratio and lowers the discovery cost, including the control overhead and storage consumption. This protocol was designed for MANETs in a fixed public area, e.g., a conference, exhibition, zoo, downtown area, or metropolis. The service discovery is based on a virtual grid framework that splits the public area into a hierarchical grid. Each node has its location information in terms of a location-aware device. Every grid cell selects a directory that is responsible for caching the service register. Some nodes (providers) provide service for other nodes (requestors). A provider registers its service along a predefined register trajectory. The service description is registered to the center line of the network (i.e., the maximum grid level). When a requestor wants to access a service, it discovers the service toward the maximum grid level. The requestor can obtain the reply from the directories. The discovery is confined to a quarter area of the network. Moreover, this study has also addressed the empty area problem for low node density. An improved register process was presented to overcome that problem. This paper has measured the performance of protocols by analysis and simulation. The analysis measured the directory ratio, register cost, and discovery cost. In our simulations, the ns2 simulator was adopted to measure the performance of each protocol. These criteria include discovery success ratio, registered directory ratio, total control packets, and discovery time. The results of our analyses and simulations demonstrated that SGrid outperforms the existing geographical service discovery protocol GCLP.

REFERENCES


Hua-Wen Tsai received the B.S. degree in information management and the M.B.A. degree in business and operations management from Chang Jung Christian University, Tainan, Taiwan, in June 1998 and June 2001, respectively, and the Ph.D. degree in computer science and information engineering from the National Cheng Kung University, Tainan, in April 2008. He joined the faculty of the Department of Computer Information and Network Engineering, Lungwha University of Science and Technology, Taoyuan, Taiwan, as an Assistant Professor in August 2008. His research interests include wireless communications, ad hoc networks, and sensor networks.

Trung-Shi Chen (M’95) received the B.S. degree in computer science and information engineering from Tamkang University, Taipei, Taiwan, in June 1989 and the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan, Taiwan, in June 1994.

He joined the faculty of the Department of Information Management, Chung Jung Christian University, Tainan, Taiwan, as an Associate Professor in June 1996, where he has been a Professor since November 2002. From June to September 2001, he was a Visiting Scholar with the Department of Computer Science, University of Illinois, Urbana-Champaign. From August 2000 to July 2003, he was the Chairman of the Department of Information Management, Chung Jung Christian University. From August 2004 to January 2008, he was a Full Professor with the Department of Information and Learning Technology, National University of Tainan, Tainan, where he was the Chairman of the Department of Information and Learning Technology from August 2004 to July 2007 and has been Full Professor with the Department of Computer Science and Information Engineering since February 2008. He was a Guest Editor of the Journal of Internet Technology and the International Journal of Ad Hoc and Ubiquitous Computing. His research interests include mobile computing and wireless networks, ubiquitous and pervasive computing, mobile learning, and data mining.

Dr. Chen is a member of the IEEE Computer Society and the IEEE Communications Society. He has served as a member of the Program Committee for several international conferences. He received the Best Paper Award at the 15th IEEE International Conference on Information Networking (ICOIN 2001) and the Third IEEE Workshop on Wireless Ad Hoc and Sensor Networks (WASN 2007).

Chih-Ping Chu (M’08) received the B.S. degree in agricultural chemistry from the National Chung Hsing University, Taichung, Taiwan, the M.S. degree in computer science from the University of California, Riverside, and the Ph.D. degree in computer science from Louisiana State University, Baton Rouge.

He is currently a Professor with the Department of Computer Science and Information Engineering, National Chung Kung University, Tainan, Taiwan. His research interests include parallelizing compilers, parallel computing, parallel processing, Internet computing, DNA computing, and software engineering.