Wireless Pers Commun
DOI 10.1007/s11277-009-9849-2

FNMGSDP: An Optimized Group-Based Service Discovery Protocol for MANETs

Zhenguo Gao · Ling Wang · Mei Yang · Jianping Wang

Abstract The ability to discover services offered in MANETs (Mobile Ad-Hoc Networks) is a major prerequisite for effective usability of MANETs. GSD (Group-based Service Discovery) protocol is a typical service discovery protocol for MANETs. However, its packet overhead is high due to its much redundant packet transmissions. Some previous works improve GSD at the expense of slightly larger cache size and packet size. However, the added information can be used to improve protocol performance further. In this paper, FNMGSDP (Forward Node Minimization enhanced Group-based Service Discovery Protocol) is proposed to minimize the number of next hop nodes when forwarding request packets by exhaustively utilize the information in Service Information Cache. Simulation results confirm the superiority of FNMGSDP over GSD and its two enhanced versions.

Keywords Service discovery protocol · MANET · NP-complete

1 Introduction

MANETs (Mobile Ad-Hoc networks) [1] are temporary infrastructure-less multi-hop wireless networks that consist of many autonomous wireless mobile nodes, and service

Z. Gao (✉)
Department of Automation, Harbin Engineering University, 150001 Harbin, China
e-mail: gag@ftcl.hit.edu.cn; gag@hrbeu.edu.cn

L. Wang
Department of Computer Science and Technology, Harbin Institute of Technology, 150001 Harbin, China
e-mail: lw@ftcl.hit.edu.cn

M. Yang
Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, NV 89119, USA
e-mail: my@egr.unlv.edu

J. Wang
Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong
e-mail: jianwang@cityu.edu.hk

Published online: 13 November 2009

discovery is a major building block of MANETs where flexibility and minimum user intervention are essential. SDP (Service Discovery Protocol) enables mobile nodes to advertise their own capabilities to the rest of the network and to automatically locate services with requested attributes. In the context of service discovery, service is any hardware or software feature that can be utilized or benefited by any node. Service description is the information that describes a service's characteristics, such as service type, attributes, access method, etc. A server is a node that provides some services. A client is a node that requests services provided by other nodes. When needing a service, a client sends out a service request packet which will be forwarded by others. When receiving a service request packet, every node with matched services should respond with a service reply packet, which will be forwarded reversely to the source of the corresponding service request packet. Other nodes without matched services should forward the service request packet. All these corresponding packet transmissions, including service request packets and service reply packets, form a SDP session.

Some service discovery protocols used in one-hop wireless networks have been proposed in [2,3]. However, they are not suitable for MANETs where multi-hop is essential.

Along with the development of MANETs, a number of service discovery protocols have been proposed. Some of them [4,5] are adapted from service discovery protocols for wired networks. Some others are designed specially for MANETs, such as [6–20]. For a detailed review of service discovery protocols for MANETs, please refer to Reference [21].

Among existing protocols, GSD [8,9] (Group-based Service Discovery protocol) is characterized by its two interesting mechanisms: (1) peer-to-peer caching of service advertisement packets and (2) group-based intelligent forwarding of service request packets. But GSD is not very efficient. Hence, several enhanced version of GSD have been proposed, such as PCPGSD (PFCN, CRN and PRN enhanced GSD protocol) [10] and CNPGSDP [11]. These works improve the efficiency of request packet forwarding greatly at the expense of a little additional information in cache. However, they can be improved further by utilizing the cached information exhaustively.

Hence, in this paper, FNMGSDP (Forward Node Minimization enhanced Group-based Service Discovery Protocol) is proposed to minimize the number of next hop nodes when forwarding request packets by making full use of the information in SIC (Service Information Cache).

The rest of the paper is organized as follows. Section 2 gives a brief review to GSD as well as its successors, PCPGSD and CNPGSDP. Section 3 describes how the number of next hop nodes when forwarding request packets is minimized in FNMGSDP with the help of nodes' local cache. Section 4 performs comparative simulation studies between FNMGSDP and other protocols including GSD, PCPGSD, and CNPGSDP. Section 5 concludes the paper.

2 Overview of GSD Related Protocols

2.1 GSD

GSD (Group-based Service Discovery protocol) was first introduced in [8], and then it was described in detail in [9]. GSD has three basic operations: (1) service advertisement packet spreading; (2) service request packet forwarding; and (3) service reply packet routing. To improve the efficiency of the three operations, GSD introduces two interesting mechanisms: (1) peer-to-peer caching of service advertisement packets; (2) group-based intelligent forwarding of service request packets.

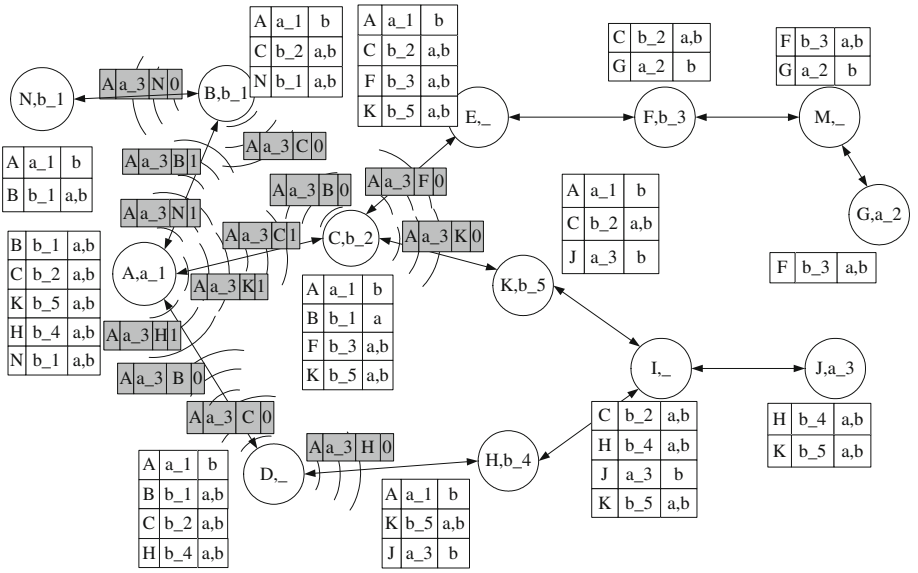


Fig. 1 Example of service request packet forwarding process in GSD

According to peer-to-peer caching of service advertisement packets, a server should generate service advertisement packets periodically, which contain the descriptions of the services provided by the server. Service advertisement packets are cached and forwarded further by receivers. The cache is called as SIC (Service Information Cache). When generating service advertisement packets, if the information in the current node's cache indicates that there are some servers in its vicinity, the group information of the services provided by these servers should also be included in the service advertisement packet. The maximum number of hops that advertisement packets can travel, denoted as b, is restricted by a user-definable parameter.

According to group-based intelligent forwarding of request packets, when an unmatched request packet should be forwarded further, the node will try to select some nodes basing on cached information. The selected nodes should have some neighbors that provide services belonging to the same group as the requested service. Such nodes are called as candidate nodes in the following text.

If the node that receives a new request packet finds matched services, it unicasts a service reply packet to the sender of the service request packet. The service reply packet will be relayed to the client of the corresponding SDP session along the reverse path.

Figure 1 shows an example of how service request packet is forwarded in GSD. In this figure, hop limits of service advertisement packets and service request packets are both set to 2; circles represent mobile nodes. The string in a circle indicates the identity of the node and the services it provides. For example, in Fig. 1, string "A, a_1" indicates that the node is A and it provides a service "a_1", which belongs to service group "a". A double-headed arrow line between two nodes indicates that these two nodes can communicate with each other directly. The white table adjacent to a node represents its SIC (not all fields are shown). For example, the 3rd item in node A's SIC, $\{H, b\_4, (a, b)\}$, indicates that: (1) this entry corresponds to node H; (2) node H provides service "b_4"; and (3) some nodes in H's 2-hop neighbor set provide group "a" services and some other nodes provide group "b" services. Arcs around a node indicate the transmission of a service request packet. The grey

table over the arcs represents the content of the packet being transmitted (not all fields are shown). For example, the grey table between nodes A and B, $\{A, a\_3, B, 1\}$, indicates that: (1) the client that initiates the SDP session is node A; (2) the requested service is "a_3"; (3) the destination of the request packet is node B; and (4) the request packet can still travel 1 hop.

When node A needs a service "a_3", which belongs to group "a", and there is no matched service in its SIC, node A has to select some nodes based on its SIC. Node A knows that the nodes B, C, H, K, and N have seen some services of group "a". Hence, five unicast request packets are sent to them, respectively. When receiving the packet, these nodes will forward the service request packet further if no match is found. For example, node B finds that: (1) no local matched services, i.e., neither services provided by node B nor services cached in SIC matches the request; and (2) the number of hops that the packet can still travel is larger than 0. Thus, node B will forward the packet further. In its SIC, node B finds that both nodes A and C have seen some group "a" services, but node A is not selected since that it is the sender of the packet. Hence, only node C is selected and a request packet is forwarded towards C. By the underlying routing protocol, the packet is sent directly to node A where it is discarded because that it is a duplicated packet. Similarly, when receiving the request packet from node A, node C forwards the packet towards nodes B, F, and K, respectively. Node D forwards the packet towards nodes B, C, and H. The request is matched at node H and K. Hence, totally 13 service request packets will be sent.

2.2 PCPGSD

Benefiting from its mechanisms, GSD achieves efficient network bandwidth usage and increased flexibility in the service matching process [8]. However, it still has several aspects that should be reconsidered.

Firstly, a copy of the request packet should be forwarded in unicast mode for each candidate node. Thus, when there are several candidate nodes, many copies of the same request packet will be sent.

Secondly, since that the maximum number of hops that a request packet can travel is restricted, there may be some candidate nodes that are too far to be reached by the request packet. However, such unreachable candidate nodes are not distinguished from common candidate nodes in GSD. Hence, packet transmissions to these unreachable candidate nodes are usually useless.

To overcome the problems, an enhanced version of GSD is proposed where three mechanisms are proposed. The three mechanisms are PFCN (Pruning of Far Candidate Nodes), CRN (Combining of Relay Nodes), and PRN (Piggybacking of Relay Nodes). Consequently, the new protocol is named as PCPGSD (PFCN, CRN and PRN enhanced GSD protocol) [10]. In PCPGSD, a node that precedes the current node on the path from the candidate node to the current node is called as relay node, or forward node. In this paper, "relay node" and "forward node" are used interchangeably.

According to PFCN, all candidate nodes that are too far to be reached by the request packet are omitted. To determine whether a candidate is too far to be reachable, a new field indicating the original maximum hop that the advertisement packet can travel is added to advertisement packets.

According to CRN, for each candidate node, a relay node should be selected and the request packet should be sent to the relay node instead of the candidate node. Obviously, several candidate nodes may share the same relay node. Thus, in such cases, instead of

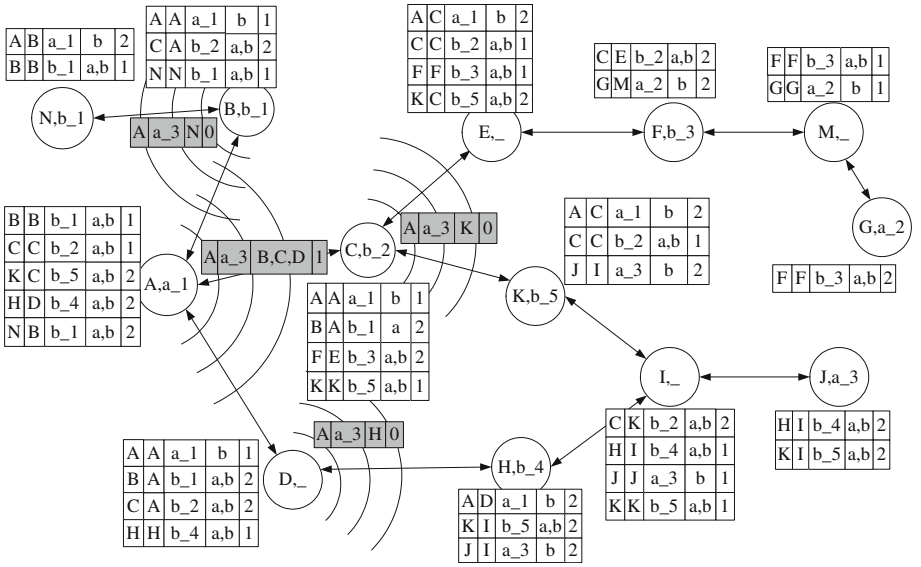


Fig. 2 Example of service request packet forwarding process in PCPGSD

sending one request packet towards each candidate nodes, only one request packet should be sent to the relay node. This single request packet is enough since that the relay node can forward the request packet intelligently basing on its local topology information. By this means, request packets are reduced further.

According to PRN, instead of sending a request packet to each relay node, a modified request packet which has a field enclosing the list of relay nodes is sent in broadcast mode, i.e., the list the relay nodes is piggybacked in the request packet. In this way, only one request packet sent in broadcast mode is enough.

Figure 2 shows an example of service request packet forwarding process in PCPGSD over the same scenario as that in Fig. 1. The new SIC field indicates the number of hops from the current node to the corresponding server, which is obtained by subtracting remain hop number of the packet from its original hop number stored in a field of it.

In Fig. 2 where PCPGSD is used, all request packets are sent in broadcast mode. Hence, although nodes B, C, H, K, and N are still candidate nodes of node A, valid receivers of the service request packet sent by node A is their corresponding relay nodes, which are nodes B, C, and D. When node B forwards the request further, it knows that nodes C and N are candidate nodes, but node C is a far candidate node. Hence, only node N is selected as a valid candidate node and a request packet is forwarded to its relay node, which is node N itself. Other nodes operate in similar way. Thus, totally 4 request packets will be sent.

2.3 CNPGSDP

CNPGSDP [11] improves PCPGSD further by reducing the number of valid candidate nodes to be considered when dispatching request packets by using a mechanism named as CNP (Candidate Node Pruning).

In CNPGSDP, CRN and PRN mechanisms of PCPGSD are combined and renamed as BSU (Broadcast Simulated Unicast) because that several unicast request packets are replaced with one request packet transmitted in broadcast mode with all unicast receivers enclosed.

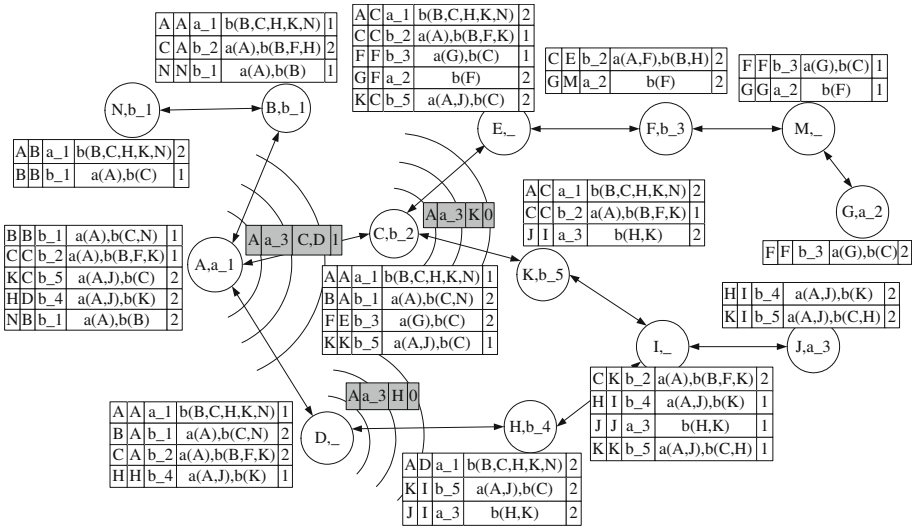


Fig. 3 Example of service request packet forwarding process in CNPGSDP

In CNPGSDP, to reduce the number of candidate nodes, candidate nodes are classified into two categories: internal candidate nodes, and external candidate nodes. If all nodes in a candidate node's d-hop vicinity that provides some services belonging to the same group as the request service are in the current node's d-hop vicinity, then the candidate node is an internal candidate node. Otherwise, the candidate node is an eternal candidate node. Reference [11] proved that all internal candidate nodes can be omitted when selecting valid candidate nodes. Hence, in CNPGSDP, all internal candidate nodes and far candidate nodes are removed from the candidate node set from which valid candidate nodes are selected. Hence, this scheme is named as CNP.

Figure 3 shows an example of service request packet forwarding process when CNPGSDP is used. Node A's 4th SIC entry is changed to $\{H, D, b\_4, (a(\mathrm{A, J}), b(\mathrm{K})), 2\}$. The enhanced field $(a(\mathrm{A, J}), b(\mathrm{K}))$ indicates that nodes A and J in H's 2-hop neighbor set provide some group "a" services, and node K provides some group "b" services. In Fig. 3, when node B is forwarding the request packet, the candidate node N will not be selected since that it is an internal node, that is, node B's SIC shows that services in group "a" seen by node N are provided by node A, whose service information is already cached by node B. Since that node B knows that the service provided by node A is "a_1", which does not match the request, the request will not be forwarded to node N. As a result, only 3 request packets will be sent.

However, in CNPGSDP, the appended information in SIC is not exhaustively used to reduce request packets. For example, in Fig. 3, both node H and K are selected as valid candidate nodes and so a request packet is broadcasted to their corresponding relay nodes D and C. But the SIC already shows that the services of group "a" seen by both node H and K are provided by nodes A and J. That means that both nodes H and K have already cached the services. Hence, a request packet sent to either of them is enough to determine whether there is a match. Considering this, FNMGSDP is proposed in this paper, which minimized the number of forward nodes when forwarding request packets by making full use of the SIC information.

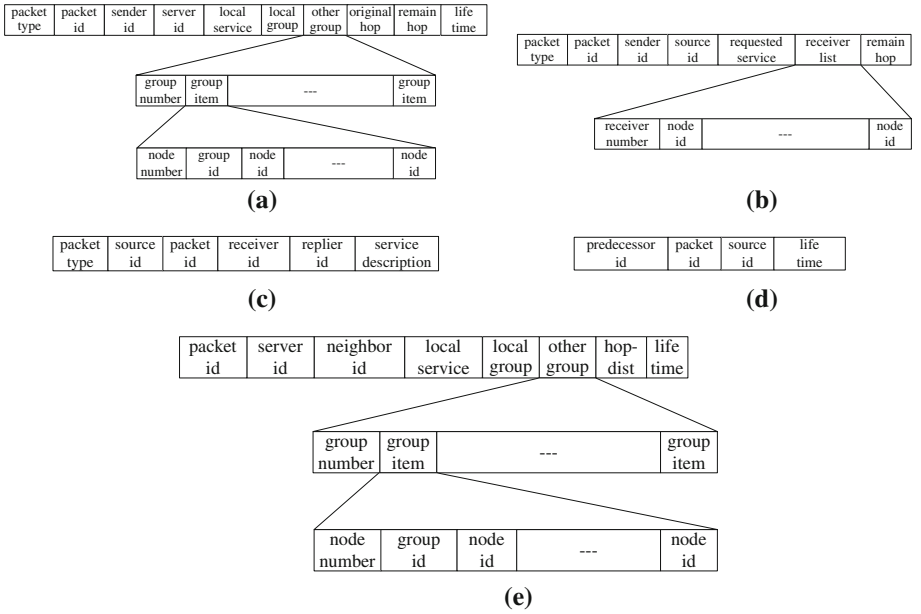


Fig. 4 Data structures in FNMGSDP. **a** structure of a service advertisement packet; **b** structure of a service request packet; **c** structure of a service reply packet; **d** structure of a RRT item; **e** structure of a SIC item

3 FNMGSDP

Packet manipulation processes and data structures of FNMGSDP are all very similar to those of CNPGSDP [11]. FNMGSDP enhances CNPGSDP mainly by using an optimized algorithm to reduce valid candidate number when forwarding request packets.

3.1 Data Structures and Packet Formats in FNMGSDP

Although most data structures and packet formats in FNMGSDP are the same to those in CNPGSDP [11], these structures are listed here for convenience.

3.1.1 Format of Service Advertisement Packet

The format of the service advertisement packet is shown in Fig. 4a. Its fields are described as follows:

packet-type:	indicates packet type, i.e., it is a service advertisement packet.
packet-id:	a number increases monotonically with each service advertisement packet generated by the node. It is used to identify different advertisement packets from the same node.
sender-id:	indicates the direct sender of the packet.
server-id:	indicates the server that generates the service advertisement packet.
local-service:	stores the description of the services provided by the server indicated by *server-id*.

service-group: stores the list of the service groups that the services in the local-service field belong to.
other-group: this compound field encloses the list of service groups that the services provided by nodes in the d-hop neighbor set of the server belong to and these servers. Each *group-item* subfield contains the *group-id* and the corresponding list of servers.
original-hop: indicates the number of hops the advertisement packet can travel, which is set by the client.
remain-hop: indicates the remaining number of hops that the packet can travel. Before forwarding the packet, the *remain-hop* field will be decreased by 1. The *remain-hop* field is initialized to a user defined value.
life-time: indicates the time period that the information in the packet can be cached in receivers' SIC.

3.1.2 Format of Service Request Packet

The format of the service request packet is shown in Fig. 4b. Its fields are described as follows:

packet-type: indicates packet type.
packet-id: a number increasing monotonically with each request packet from a client.
sender-id: indicates the direct sender of the packet.
source-id: indicates the node that generates the request packet. A pair (*source-id*, *packet-id*) uniquely identifies a SDP session.
receiver-list: The *receiver-list* compound field stores the list of receivers selected by the sender. Its *receiver-number* subfield indicates the number of receivers in the list.
request-description: stores the description the requested service.
remain-hop: indicates the number of hops that the packet can still travel. If this field is 0, the packet will be dropped.

3.1.3 Format of Service Reply Packet

The format of the service reply packet is shown in Fig. 4c. Its fields are described as follows:

packet-type: indicates packet type.
source-id: indicates the node that generates the corresponding request packet.
packet-id: the value of the *packet-id* field of the corresponding request packet
receiver-id: indicates the next-hop node of the reply packet.
replier-id: indicates the node that generates the reply packet.
service-description: stores the description of the matched services.

3.1.4 Structure of RRT

Each node maintains a RRT (Reverse Route Table), which is used in two tasks: (1) checking duplicated request packets, and (2) routing service reply packets to the corresponding source node. Fig. 4d shows the structure of an RRT entry. The *predecessor-id* field indicates the

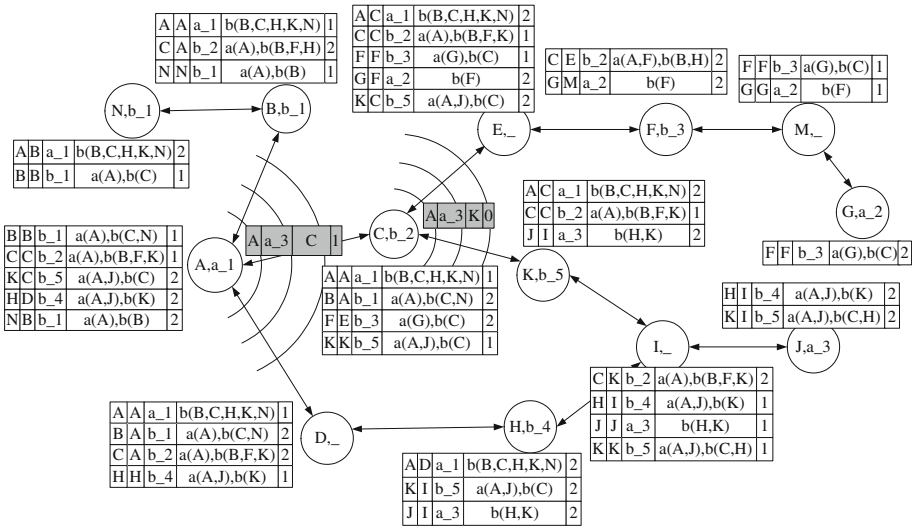


Fig. 5 Example of service request packet forwarding process in FNMGSDP

node from which the request packet is received. The *packet-id* field and the *source-id* field are the same to those of a request packet.

3.1.5 Structure of SIC

SIC is used to cache service advertisement packets. Structure of SIC is shown in Fig. 4e. All fields are the same as those of the service advertisement packet except for: (1) the *neighbor-id* field indicates the node from which the service advertisement packet is received; (2) the *hop-dist* field indicates the number of hops from the current node to the corresponding server who originates the advertisement packet, which is obtained by subtracting the value of the remain-hop field from the value of the original-hop field of the corresponding advertisement packet.

3.2 Preliminaries of Request Packet Forwarding Algorithm in FNMGSDP

In the following text, some preliminaries of the request packet forwarding algorithm in FNMGSDP are described with the scenario shown in Fig. 5 as an example. The example shows the forwarding process of a SDP session searching for service "*a*_3" originated at node *A*. In Fig. 5, hop limits of service advertisement packets and service request packets are set to 2. Meanings of the symbols in the figure are just the same to those in Figs. 1, 2, and 3. Detailed processes of forward nodes selection performed by nodes *A* and *C* are shown in Table 1.

3.2.1 Symbols

Table 1 lists some symbols used in the description of FNMGSDP in the following text. To facilitate the understanding of the symbols, some examples with node *C* is the current node are given in the corresponding 3rd column.

Table 1 List of symbols

Symbol	Meaning	Examples
c	The client that generates the current service request	A
u	The current node	C
t	The sender of the service request packet received by u	A
d	The maximum number of hops that advertisement packets can travel	2
g	The group that the requested service belongs to	a
$R(t)$	The set of nodes stored in the *receiver-list* field of the service request packet sent by node t. These nodes should forward the service request packet. At the service request packet's client $u, t = u$, $R(t) = \{u\}$	C
$N_x(u)$	The set of nodes that are at most x-hop away from node u. It is node u's x-hop neighbor set (excluding node u itself)	$N_2(C) = \{A, B, D, E, F, H, I, K\}$
$e(u, s)$	The entry that corresponds to server s in node u's SIC	$e(C, F)$ corresponds to the 3rd item of node C's SIC
$E(u)$	The set of all entries in node u's SIC	$E(C) = \{e(C, A), e(C, B), e(C, F), e(C, K)\}$
$H_c(u, s, g)$	The set of nodes in $N_d(s)$ that provide some services belonging to group g, where node s is a server in $N_d(u)$. In other words, $H(u, s, g)$ is the set of nodes in the *group-item* field in $e(u, s)$'s *other-group* field whose *group-id* field is g	$H_{\mathrm{C}}(C, F, a) = \{G\}$ $H_{\mathrm{C}}(C, K, a) = \{A, J\}$
$S(u)$	The set of servers in $N_d(u)$	$\mathrm{S}(C) = \{A, B, F, K\}$
$C(u, g)$	$C(u, g) = \{s \mid H_{\mathrm{c}}(u, s, g) \neq \emptyset, s \in S(u)\}$	$C(C, a) = \{B, F, K\}$
$f(u, s)$	The node indicated by the *neighbor-id* field of the entry corresponding to node s in u's SIC. It is the next-hop node on the path from node u to s	$f(C, A) = A$; $f(C, B) = A$; $f(C, F) = E$; $f(C, K) = K$;
$F(u, g)$	$F(u, g) = \{f(u, s) \mid s \in C(u, g)\}$	$F(C, a) = \{f(u, s) \mid s \in C(C, a)\} = \{A, E, K\}$
$C_f(u, f_0, g)$	$C_{\mathrm{f}}(u, f_0, g) = \{s \mid s \in C(u, g), f(u, s) = f_0\}$	$C_{\mathrm{f}}(C, A, a) = \{s \mid s \in C(C, a), f(C, s) = A\} = \{A, B\}$
$C_R(u, g)$	$C_{\mathrm{R}}(u, g) = \{s \mid s \in C(u, g), e(u, s).hop - dist > V_{\mathrm{remain-hop}}\}$ Here $V_{remain-hop}$ is the value of the request packet's *remain-hop* field	$C_{\mathrm{R}}(C, a) = \{s \mid s \in C(C, a), e(u, s).hop - dist > 1\} = \{B, F\}$
$H_{ALL}(u, g)$	$H_{\mathrm{ALL}}(u, g) = \bigcup_{s \in C(u,g)} H_{\mathrm{c}}(u, s, g)$	$H_{\mathrm{ALL}}(C, a) = \{A, G, J\}$
$H_f(u, f_0, g)$	$H_{\mathrm{f}}(u, f_0, g) = \bigcup_{s \in C_{\mathrm{f}}(u, f_0, g)} H_{\mathrm{c}}(u, s, g)$	$H_{\mathrm{f}}(C, A, a) = \{A\}$ $H_{\mathrm{f}}(C, E, a) = \{G\}$ $H_{\mathrm{f}}(C, K, a) = \{A, J\}$

3.2.2 Definitions

Definition 1: (*Candidate Node*) All nodes in $C(u, g)$ are called as Candidate Nodes of node u.
Definition 2: (*Forward Node*) The node indicated by $f(u, s)$ is called as the Forward Node corresponding to server s relative to the current node u.
Definition 3: (*Far Candidate Node*) All candidate nodes in $C_R(u, g)$ are called as Far Candidate Node.
Definition 4: (*Hidden Server*) Each node in $H_c(u, s, g)$ is called as a hidden server of server s relative to the current node u.
Definition 5: (*Extended Coverage*) If node c receives a request packet of a SDP session, then each server s in $N_d(c)$ is in the extended-coverage of the SDP session. In this case, server s is said extendedly-covered by the SDP session, or s is extendedly-covered by node c.
Defnition 6: (*Virtual Coverage*) If node u sends a request packet towards a candidate node c, then we said that all hidden servers in $H_c(u, c, g)$ are in the virtual-coverage of the current node u. We can also said that servers in $H_c(u, c, g)$ are all virtually-covered by node u.
Definition 7: (*Coverage-Needed Hidden Server*) When making request packet forwarding decisions, not all hidden servers are necessary to be virtual covered by the current node, because that the current node may be sure basing on locally available information that some hidden servers are not necessary to be considered. A hidden server that should be virtually-covered by the node u is called as a coverage-needed hidden server of a node u.

For example, when node C making request packet forwarding decisions in Fig. 5, it knows that as the client of the current session, node A must not know any matched services. Hence, although node A and J are both hidden servers of node C, the hidden server A is not its Coverage-Needed Hidden Server. The set of all Coverage-Needed Hidden Servers of node C is $\{J\}$.

We denote the set of coverage-needed hidden servers of the current node u as $H_{TC}(u, g)$. Obviously, $H_{\mathrm{TC}}(u, g) \subseteq H_{\mathrm{ALL}}(u, g)$.

From the definition of Coverage-Needed Hidden Server, so long as the current node could guarantee that all its Coverage-Needed Hidden Servers are in the virtual coverage of the current node, then all hidden servers of the current node are guaranteed in the current service request session.

Defnition 8: *DFNS* (*Dominating Forward Node Set*) When forwarding request packets, a *DFNS* of the current node u is a set of forward nodes guaranteeing that all hidden servers seen by the current node u could be extendedly-covered by the current SDP session. A *DFNS* of the current node u is denoted as $F_{DFNS}(u, g)$ in the following text.

3.2.3 Find Minimum Dominating Forward Node Set

Each forward node will have to forward the service request packet, unless that it found matched services or hop-limit is reached. Hence, in order to reduce request packet overhead, the size of DFNS should be minimized. The task of finding a DFNS with minimum size is called as DFNS problem.

Using H to represent the set of coverage-needed hidden servers $H_{TC}(u, g)$, C to represent the family of sets $\{H_f(u, f_0, g) | f_0 \in F(u, g)\}$, and F to represent the set of all forward nodes $F(u, g)$, the DFNS problem can be defined as follows:

DFNS problem: Given H, C, and F, find a dominating forward node set $F_{DFNS}(u, g) \subseteq F$ with minimum size.

Since $\cup_{f_0 \in C} H_f(u, f_0, g) = H_{ALL}(u, g) \supseteq H$ and there is a 1-to-1 correspondence between C and F, the decision version DFNS Problem can be defined formally as follows:

DFNS problem (decision version): Given a positive integer k, $H = \{h_1, \ldots, h_n\}$, $C = \{C_1, \ldots, C_m\}$, and $\cup_{C_i \in C} C_i \supseteq H$, decides whether there is subset $B \subseteq C$ with size k such that $\cup_{C_i \in B} C_i \supseteq H$.

Theorem 1 *The DFNS problem is NP-complete.*

Proof First, we show that the DFNS problem belongs to NP. Suppose we are given a solution $F_{DFNS}(u, g)$ and an integer k. The size of $F_{DFNS}(u, g)$ can be verified in $O(1)$. Whether $F_{DFNS}(u, g)$ is a dominating forward node set can be verified in $O(n^2)$, where n is the maximum number of hidden servers of a node. Hence, the total time needed for verification is in polynomial time.

Now we show the DFNS problem is NP-hard by showing that a well-known NP-complete problem, the Set Cover (SC) problem [22], is polynomially reducible to the DFNS problem, i.e., SC $\leq_p$ DFNS.

The definition of the SC Problem is as follows:

Set Cover: Given $U = \{u_1, \ldots, u_n\}$, $S = \{S_1, \ldots, S_m\}$, $S_i \subseteq U (i = 1, \ldots, m)$, and $\cup_{S_i \in S} S_i = U$, a subset $S' \subseteq S$ is a set cover of$U$ if $\cup_{S_i \in S'} S_i = U$.

Set Cover problem (decision version): Given an positive integer k', $U = \{u_1, \ldots, u_n\}$, $S = \{S_1, \ldots, S_m\}$, $S_i \subseteq U (i = 1, \ldots, m)$ and $\cup_{S_i \in S} S_i = U$, determines whether there is a set cover $S' \subseteq S$ with size k' such that $\cup_{S_i \in S'} S_i = U$.

With the help of $T' = \{T'_1, \ldots, T'_m\}$, $(\cup_{T'_i \in V'} T'_i) \cap U = \Phi$, the reduction function f converts U, S and k' of SC problem to H, C and k of DFNS problem as follows, where $k = k'$,

$$H = \{h_1, h_2, \ldots, h_n\} = \{u_1, u_2, \ldots, u_n\} = U,$$
$$C = \{C_1, C_2, \ldots, C_m\} = \{S_1 \cup T'_1, S_2 \cup T'_2, \ldots, S_m \cup T'_m\}.$$

Clearly, f can be performed in polynomial time as H and C can be obtained in $O(n)$ time.

Then we show that f is a reduction function: a DFNS of size k can be found in (H, C) if and only if U has a set cover of size k'.

We first show that if U has a set cover of size k', (H, C) must have a DFNS of size k. Assume that S' is a set cover of U with size k', and $V = \{i | S_{i \in} S'\}$, then,

$$\bigcup_{i \in V} C_i = \bigcup_{i \in V} (S_i \cup S'_i) = \left(\bigcup_{i \in V} S_i\right) \cup \left(\bigcup_{i \in V} S'_i\right) \supseteq \bigcup_{i \in V} S_i = U = H.$$

Hence, $C' = \{C_i | i \in V\}$ is a DFNS of (H, C). Moreover, its size is $k = k'$.

Conversely, if (H, C) has a DFNS of size k, U must have a set cover of size k'. Assume that C' is a DFNS of size k and $V' = \{i | C_i \in C'\}$, then we have:

$$\begin{aligned} U = H &= \left(\bigcup_{i \in V'} C_i\right) \cap H = \left(\bigcup_{i \in V'} (S_i \cup S_i')\right) \cap H \\ &= \left(\left(\bigcup_{i \in V'} S_i\right) \cap H\right) \cup \left(\left(\bigcup_{i \in V'} S_i'\right) \cap H\right) \\ &= \left(\left(\bigcup_{i \in V'} S_i\right) \cap U\right) \cup \left(\left(\bigcup_{i \in V'} S_i'\right) \cap U\right) \\ &= \left(\bigcup_{i \in V'} S_i \cap U\right) \cup \Phi \\ &= \left(\bigcup_{i \in V'} S_i\right) \cap U \end{aligned}$$

As such, we have

$$\bigcup_{i \in V'} S_i \supseteq U \tag{1}$$

According to the definition of SC problem, we have

$$\bigcup_{i \in V'} S_i \subseteq \bigcup_{i \in \{1, \ldots, m)} S_i = U \tag{2}$$

Combining Eqs. (1) and (2), we get

$$\bigcup_{i \in V'} S_i = U.$$

Thus, $S' = \{S_i | i \in V'\}$ is a set cover of U, and its size is $k' = k$.

Therefore, the DFNS problem is NP-Complete. □

Since that DFNS is an NP-complete problem, the following greedy heuristics is proposed in FNMGSDP to select a dominating forward node set $F_{DFNS}(u, g)$ from H, C, and F.

Algorithm *Greedy Minimum DFNS Algorithm*

1. Let $F_{DFNS}(u, g) = \Phi$ (empty set), $H_{RC}(u, g) = \Phi$.
2. Find node f_0 among $F(u, g)$ with the maximum $|(H_f(u, f_0, g) \cap H_{TC}(u, g)) \setminus H_{RC}(u, g)|$. In case of a tie, select f_0 with maximum $|H_f(u, f_0, g) \cap H_{TC}(u, g)|$. In case of a new tie, select f_0 with smallest ID.
3. $F_{DFNS}(u, g) = F_{DFNS}(u, g) + f_0$, $F(u, g) = F(u, g) - f_0$,
4. $H_{RC}(u, g) = H_{RC}(u, g) \cup H_f(u, f_0, g)$. If $H_{RC}(u, g) \supseteq H_{TC}(u, g)$, exit; Otherwise, go to Step 2.

Corollary 1 *The approximation ratio of the greedy Minimum DFNS algorithm for DFNS problem is* $ln|H| + 1$ ($|H|$ *means the size of set* H).

Proof The approximation ratio of the similar greedy heuristics for SC problem is $ln|H|+1$ [22].

Since DFNS problem is intrinsically a SC problem, the approximation ratio of our algorithm for DFNS problem is also $ln|H|+1$. □

3.3 Service Request Forwarding Algorithm in FNMGSDP

When receiving a new service request packet, each node that knows about some matched services should respond with a service reply packet. Otherwise, it forwards the service request packet if either of the following two conditions is matched:

- The *receiver-number* field of the packet's *receiver-list* field is 0.
- The *receiver-number* field of the packet's *receiver-list* field is greater than 0 and the current node is in the *receiver-list* field.

If the current node determines to forward the service request packet, it will perform the following algorithm.

Algorithm *Service Request Packet Forwarding*

1. Determines the set of coverage-needed hidden servers using Eq. (3).
2. Calls the greedy Minimum DFNS algorithm to obtain a dominating forward node set $F_{DFNS}(u,g)$ from $H_{TC}(u,g)$, $F(u,g)$, and $\{H_f(u,f_0,g)|f_0 \in F(u,g)\}$.
3. Encloses nodes in $F_{DFNS}(u,g)$ into the service request packet's *receiver-list* compound field, and set the *receiver-number* sub-field of the *receiver-list* field to the number of nodes in $F_{DFNS}(u,g)$.
4. Decreases the *remain-hop* field by 1.
5. Sends out the service request packet in broadcast mode.

$$H_{TC}(u,g) = \bigcup_{s\in C(u,g)\setminus C_R(u,g)} H_c(u,s,g) - R(t)\cup\{c,t,u\}\cup C(u,g) - \bigcup_{s\in(R(t)\cup\{c,t,u\})\cap C(u,g)} H_c(u,s,g) \quad (3)$$

3.4 Determining the Set of Coverage-Needed Hidden Servers

In the service request packet forwarding algorithm used in FNMGSDP, the set of $H_{TC}(u,g)$ is determined according to Eq. (3). The correctness of excluding $R(t)\cup\{t,u\}\cup C(u,g)$ and $\bigcup_{s\in(R(t)\cup\{t,u\})\cap C(u,g)} H_c(u,s,g)$ from $H_{TC}(u,g)$ when making request packet forwarding decisions is shown in the following Lemmas 1 and 2 and Theorem 2. They are based on the assumption that packet transmissions are error free.

Lemma 1 *$R(t)\cup\{c,t,u\}\cup C(u,g)$ can be removed from $H_{TC}(u,g)$.*

Proof It will be proved by showing that for each node $w \in R(t)\cup\{c,t,u\}\cup C(u,g)$, w can be excluded from $H_{TC}(u,g)$. There are three cases:

- If $w\in\{c,t,u\}$, then w has received the request packet no matter w is c, t or u. Hence, w has already been covered by the SDP session. In other words, it has already been extendedly-covered by the SDP session. Therefore, w can be excluded from $H_{TC}(u,g)$.

- If $w \in R(t)$, then w is another valid receiver of the current request packet. Thus, w must be able to receive the packet also. Hence, w can be covered by the SDP session. Hence, w can be excluded from $H_{TC}(u, g)$.
- If $w \in C(u, g) - R(t) - \{c, t, u\}$, then since $w \in C(u, g)$, there must be a entry corresponding to node w in node u's SIC. Thus, w is extendedly-covered by the current SDP session. Hence, w can be excluded from $H_{TC}(u, g)$.

Hence, the lemma follows. □

Lemma 2 $\bigcup_{s \in (R(t) \cup \{c,t,u\}) \cap C(u,\, g)} H_c(u, s, g)$ *can be removed from* $H_{TC}(u, g)$.

Proof It will be proved by showing that for each node $w \in (R(t) \cup \{c, t, u\}) \cap C(u, g)$, $H_C(u, w, g)$ can be excluded from $H_{TC}(u, g)$.

Since $w \in (R(t) \cup \{c, t, u\}) \cap C(u, g)$, $w \in R(t) \cup \{c, t, u\}$ and $w \in C(u, g)$.

- Since $w \in R(t) \cup \{c, t, u\}$, w must be able to receive a request packet of the SDP session.
- Since $w \in C(u, g)$, then for each w, there is a corresponding entry in the current node u's SIC. Hence, all hidden servers in $Hc(u, w, g)$ have already been extendedly-covered by the current SDP session. Thus, they can be removed from $H_{TC}(u, g)$.

Hence, the lemma follows. □

Theorem 2 *The set of coverage-needed hidden servers of current node u can be expressed as:*

$$H_{TC}(u, g) = \bigcup_{s \in C(u,g) \setminus C_R(u,g)} H_c(u, s, g) - R(t) \cup \{c, t, u\} \cup C(u, g) - \bigcup_{s \in (R(t) \cup \{c,t,u\}) \cap C(u,g)} H_c(u, s, g)$$

Proof The set of all hidden servers seen by the current node u is $H_{ALL}(u, g)$. However, considering the hop limit of service request packets, Far Candidate Nodes in $C_R(u, g)$ are unreachable for the request packet from the current node u. Thus, the set of hidden servers that could be virtually-covered by the current node u can be expressed as:

$$H_{CC}(u, g) = \bigcup_{s \in C(u,g) \setminus C_R(u,g)} H_c(u, s, g).$$

Additionally, some hidden servers can be removed.

- According to Lemma 1, $R(t) \cup \{c, t, u\} \cup S(u, g)$ can be removed.
- According to Lemma 2, $\bigcup_{s \in (R(t) \cup \{c,t,u\}) \cap C(u,g)} H_c(u, s, g)$ can be removed.

Thus, the expression of H_{TC}(u, g) can be obtained easily as shown in this theorem. □

In FNMGSDP, the size of dominating forward node set is minimized. Hence, this scheme is called as FNM (Forward Node Minimization). Correspondingly, the protocol is called as FNMGSDP (Forward Node Minimization enhanced Group-based Service Discovery Protocol).

FNMGSDP tries to reduce request packet overhead at the expense of computation complexity. But considering that packet transmission consumes the main part of power consumption of wireless nodes, and power energy is more critical than computing resource and memory resource, the approach of FNMGSDP to enhance protocol performance is acceptable.

Table 2 Forward node selection process of nodes A and C

Process	Selection process of node A	Selection process of node C
$u, c, t, R(t)$	$u = A, c = A, t = A, R(t) = \{A\}$	$u = C, c = A, t = A, R(t) = \{C\}$
$S(u)$	$S(A) = \{B, C, H, K, N\}$	$S(C) = \{A, B, F, K\}$
$C(u, g)$	$C(A, g) = \{B, C, H, K, N\}$	$C(C, g) = \{B, F, K\}$
$C_R(u, g)$	$C_R(A, g) = \{\}$	$C_R(C, g) = \{B, F\}$
$F(u, g)$	$f(A, B) = B, f(A, C) = C,$ $f(A, H) = D, f(A, K) = C,$ $f(A, N) = B$	$f(C, B) = A, f(C, F) = E, f(C, K) = K$
$C_f(u, f_0, g)$	$C_f(A, C, g) = \{s \mid s \in C(A, g),$ $f(A, s) = C\}\{C, K\}$ $C_f(A, B, g) = \{s \mid s \in C(A, g),$ $f(A, s) = B\} = \{B, N\}$	$C_f(C, A, g) = \{s \mid s \in C(C, g),$ $f(C, s) = A\} = \{B\}$ $C_f(C, E, g) = \{s \mid s \in C(C, g),$ $f(C, s) = E\} = \{F\}$ $C_f(C, K, g) = \{s \mid s \in C(C, g),$ $f(C, s) = K\} = \{K\}$
$F(u, g)$	$F(A, g) = \{B, C\}$	$F(C, g) = \{A, E, K\}$
$H_c(u, s, g)$	$H_c(A, B, g) = \{A\},$ $H_c(A, C, g) = \{A\},$ $H_c(A, N, g) = \{A\},$ $H_c(A, H, g) = \{A, J\},$ $H_c(A, K, g) = \{A, J\}$	$H_c(C, B, g) = \{A\},$ $H_c(C, F, g) = \{G\},$ $H_c(C, K, g) = \{A, J\},$
$H_f(u, f_0, g)$	$H_f(A, B, g) = \{A\},$ $H_f(A, C, g) = \bigcup_{s \in C_f(A,C,g)} H_c(A, s, g)$ $= \{A\} \cup \{A, J\} = \{A, J\}$	$H_f(C, A, g) = \{B\},$ $H_f(C, E, g) = \{F\},$ $H_f(C, K, g) = \{K\},$
$H_{TC}(u, g)$	$H_{TC}(A, g) = \{J\}$	$H_{TC}(C, g) = \{J\}$
$F_{DFNS}(u, g)$	$F_{DFNS}(A, g) = \{C\}$	$F_{DFNS}(C, g) = \{K\}$
Request packet	$\{A, \text{“}a\_3\text{”}, C, 1\}$	$\{A, \text{“}a\_3\text{”}, K, 0\}$

3.5 Example of Spreading Process a Service Discovery Session in FNMGSDP

Detailed spreading process of the service discovery session in Fig. 5 where FNMGSDP is used is shown in Table 2.

Since that no local services match the request, node A selects its forward nodes following the process shown in the second column in Table 2. At last, only one forward node C is selected. Correspondingly, the request packet with content $\{A, \text{“}a\_3\text{”}, C, 1\}$ is sent in broadcast mode.

Similarly, when receiving the service request packet sent by node A, node C selects its forward nodes following the process as shown in the third column in Table 2. Hence, only one forward node K is selected and the request packet with content $\{A, \text{“}a\_3\text{”}, K, 0\}$ is sent in broadcast mode.

When receiving the service request packet sent by node C, Node K finds from its SIC that node J provides a service that matches the request. Hence, node H sends out a service reply packet which will arrive at the source node A along the path K-C-A.

$H_{TC}(A, g)$ and $H_{TC}(C, g)$ are obtained individually as follows:

$$H_{TC}(A, g) = \bigcup_{s \in C(u,g) \setminus C_R(u,g)} H_c(u, s, g) - R(t) \cup \{c, t, u\} \cup C(u, g) - \bigcup_{s \in (R(t) \cup \{c,t,u\}) \cap C(u,g)} H_c(u, s, g)$$

$$
\begin{aligned}
&= \bigcup_{s \in C(A,g) \setminus C_R(A,g)} H_c(A,s,g) - \{A\} \cup \{A, A, A\} \cup C(A,g) \\
&\quad - \bigcup_{s \in (\{A\} \cup \{A,A,A\}) \cap C(A,g)} H_c(A,s,g) \\
&= \bigcup_{s \in \{B,C,H,K,N\}} H_c(A,s,g) - \{A\} \cup \{A, A, A\} \cup \{B, C, H, K, N\} \\
&\quad - \bigcup_{s \in (\{A\} \cup \{A,A,A\}) \cap \{B,C,H,K,N\}} H_c(A,s,g) \\
&= \{A, J\} - \{A, B, C, H, K, N\} - \{\} = \{J\} \\
H_{TC}(C,g) &= \bigcup_{s \in C(u,g) \setminus C_R(u,g)} H_c(u,s,g) - R(t) \cup \{c,t,u\} \cup C(u,g) \\
&\quad - \bigcup_{s \in (R(t) \cup \{c,t,u\}) \cap C(u,g)} H_c(u,s,g) \\
&= \bigcup_{s \in C(C,g) \setminus C_R(C,g)} H_c(C,s,g) - \{C\} \cup \{A, A, C\} \cup C(C,g) \\
&\quad - \bigcup_{s \in (\{C\} \cup \{A,A,C\}) \cap C(C,g)} H_c(C,s,g) \\
&= \bigcup_{s \in \{K\}} H_c(C,s,g) - \{C\} \cup \{A, A, C\} \cup \{B, F, K\} \\
&\quad - \bigcup_{s \in (\{C\} \cup \{A,A,C\}) \cap \{B,F,K\}} H_c(C,s,g) \\
&= \{A, J\} - \{A, B, C, F, K\} - \{\} = \{J\}
\end{aligned}
$$

4 Performance Simulations

4.1 Select Service Discovery Protocols to Be Tested

To verify the effectivity of the improvements made in FNMGSDP, FNMGSDP and its predecessors GSD, PCPGSD, CNPGSDP are all included in our comparative simulation studies. Additionally, flood is used as the benchmark of our simulation analysis. Besides above selected schemes, a theoretically optimized scheme that searching the network using the breadth-first scheme is used to show the most ideal performance of service discovery protocols. This optimized scheme is designated as IDEAL in the following text. Thus, the schemes examined in our simulation study are: IDEAL, FLOOD, GSD, PCPGSD, CNPGSDP, and FNMGSDP.

4.2 Performance Metrics

The performance metrics considered in our simulations include: (1) number of service discovery packets, which is averaged over all service discovery sessions except for self matched sessions where matched services are found in the client's local cache. This metric is the sum of the following two metrics; (2) number of request packets, which is also averaged over

Table 3 Basic parameters

Parameters	Value
Scenario	1,000 × 1,000 m
Node number	100
Simulation time	1,000 s
Wireless bandwidth	1(Mbps)
SDP session number	100
Service advertisement interval	20 (s)
Valid time of SIC item	21 (s)
Number of servers	50
Maximum hop of advertisement packets	2
Number of service group	2
Number of service info in each group	5
Maximum hop of request packets	3
Node speed max	10 (m/s)
Node speed min	0 (m/s)
RWP pause time(s)	100 (s)

all service discovery sessions; (3) number of reply packets, which is also averaged over all service discovery sessions; (4) percent of succeeded service discovery sessions, which reflects the effectiveness of service discovery protocols; (5) Response time, which is the interval between the arrival of the first reply packet and the origination of the corresponding request packet. This metric is averaged over all succeeded but not self-matched SDP sessions. It measures the promptness of service discovery protocols.

The enhancements of FNMGSDP over its predecessors can be best understood by studying the number of candidate nodes of different categories, such as Far Candidate Nodes, Internal Candidate nodes, External Candidate nodes, etc. Hence, the corresponding data are also collected and shown in the following text.

4.3 Simulation Settings

Simulation studies are performed using GloMoSim [23]. The DCF (Distributed Coordination Function) of IEEE 802.11 is used as the underlying MAC protocol. Random waypoint model is used as the mobility model. In this model, nodes move towards their destinations with a randomly selected constant speed V. When reaching its destination, a node keeps static for a random period T_P. When the period expires, the node randomly selects a new destination and moves to the new destination with a new speed. The process will repeat permanently. In our simulations, T_P is fixed at 0.

Some basic parameters used in all these simulations are shown in Table 3. At the beginning of each simulation, 100 nodes are randomly distributed in the scenario, and a predetermined number of nodes are randomly selected as servers which provide randomly selected services. During each simulation, 100 SDP sessions are started at randomly selected time by randomly selected clients. For each simulation experiment, 50 simulations were performed. Simulation results shown in the following section are all averaged over 50 simulations. Error bars in all the following figures report 95% confidence.

4.4 Simulation Results

4.4.1 Effects of Radio Range

To inspect the effect of radio range on protocol performances, three simulation sets are performed. The radio ranges in these simulation sets are set to 100, 150, and 200 m, respectively. The results are shown in Fig. 6.

Under different radio range, FNMGSDP has the lowest service discovery packet overhead (Fig. 6a). When radio range is 200m, average numbers of service discovery packets for each service discovery session for GSD, PCPGSD, CNPGSDP, and FNMGSDP are about 64, 20, 14, and 9, respectively. FNMGSDP's lowest service discovery packet overhead mainly resulted from its least request packet overhead (Fig. 6b), which makes up the main body of service discovery packets. RNMGSDP is not the protocol that has the least reply packet overhead (Fig. 6c). However, since that reply packets are quite fewer than request packets, service discovery packets in FNMGSDP are still fewer than other protocols. There are no distinctive differences between FNMGSDP and its predecessors in terms of percent of succeeded service discovery sessions (Fig. 6d). Service discovery response in FNMGSDP is quicker than other protocols except for IDEAL as shown in Fig. 6e. This is due to the service advertisement packet spreading operation, which leads to the situation that all nodes in a server's d-hop neighbor set can see the services provided by the server. Thus, a service request will be matched in fewer hops. In our implementation, packet transmission time is not simulated in IDEAL, hence response time of IDEAL is 0s.

The superiority of FNMGSDP over its predecessors can be better explained by the numbers of candidate nodes of different categories, as shown in Fig. 6f. In this figure, the curve designated as "Candidate" represents the average number of candidate nodes of a node when forwarding a service request packet. The curves designated as "FarCandidate", "IntCandidate", "ExtCandidate" represents the average number of far candidate nodes, internal candidate nodes, and external candidate nodes, respectively. The curve designated as "Forward" represents the average number of forward nodes selected by a node when forwarding service request packets. The curve designated as "DFNSNode" represents the number of DFNS-Nodes selected by FNMGSDP protocol. As radio range increases, the number of candidate nodes and external candidate nodes increase correspondingly. When radio range is 200 m, the number of candidate nodes, far candidate nodes, internal candidate nodes, external candidate nodes, forward nodes, and DFNS nodes are 18.51, 3.2, 2.7, 12.6, 7.0, and 2.2, respectively.

4.4.2 Effects of Maximum Hop of Service Advertisement Packets

To inspect the effect of maximum hop of service advertisement packets on protocol performances, four additional simulation sets are performed. The maximum hop of service advertisement packet in the simulation sets are set to 1, 2, 3, and 4, respectively. In all these simulations, radio range is fixed to 150 m. The results are shown in Fig. 7.

Figure 7b shows that, as maximum hop that service advertisement packets can travel increases, service request packet overhead of GSD increase sharply, while in other GSD based protocols especially FNMGSDP, this metric drops down. This is easy to explain. As the maximum hop increases, service description of each service tends to be cached at more nodes. Hence, when forwarding service request packets, more candidate nodes will be found. Since that, In GSD, a request packet will be sent to each candidate node, the number of request packets will increase greatly. However, in other GSD-based protocols, BSU is used to substitute one request packet for these unicast packets in GSD, which eliminates the

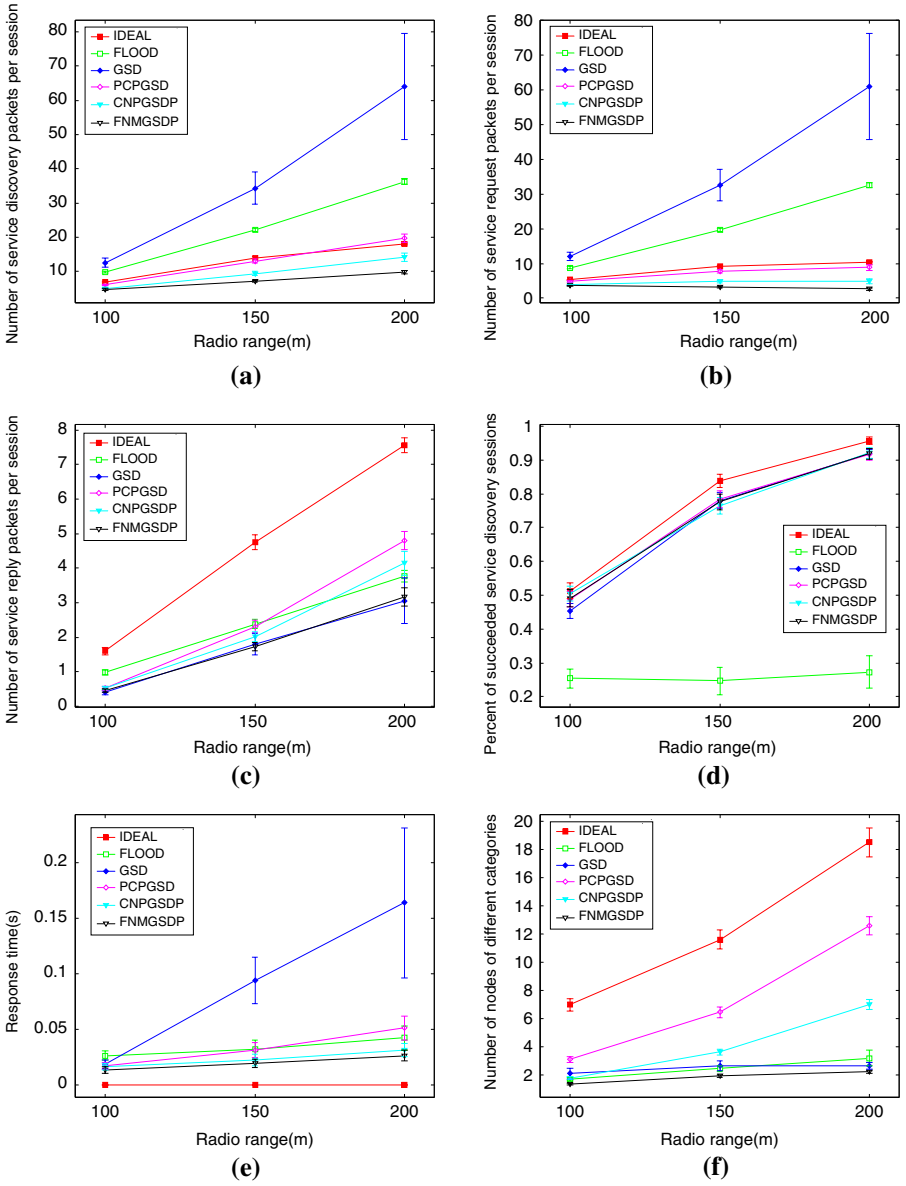


Fig. 6 Effects of radio range. **a** number of service discovery packets per session; **b** number of service request packets per session; **c** number of service reply packets per session; **d** percent of succeeded service discovery sessions; **e** response time; **f** number of candidate nodes of different categories

negative effect of more candidate nodes. Along with other effective schemes, the request packet overhead in FNMGSDP decreases greatly. The result in Fig. 7e confirms the explanation. As the maximum hop of service advertisement packets increases, the number of candidate nodes and far candidate nodes increase sharply. When maximum hop is 4, the number of candidate nodes, far candidate nodes, internal candidate nodes, external candidate nodes, forward nodes, and DFNS nodes are 21, 11.7, 1.9, 7.3, 3.4, and 1.5, respectively. Since

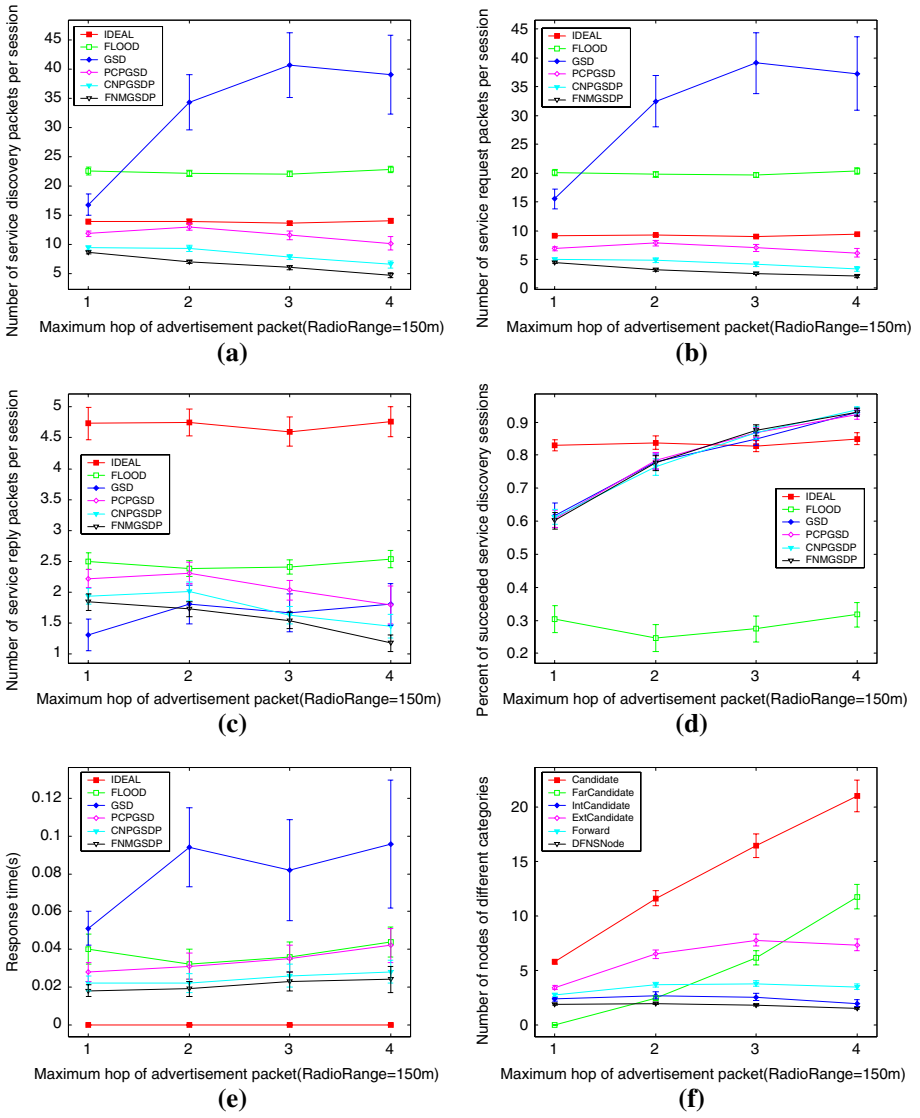


Fig. 7 Effects of maximum hop of service advertisement packets. **a** number of service discovery packets per session **b** number of service request packets per session; **c** number of service reply packets per session; **d** percent of succeeded service discovery sessions; **e** response time; **f** number of candidate nodes of different categories

that service request packets make up a high ratio in service discovery packets, the curves of service discovery packet overhead in Fig. 7a show similar trend as those in Fig. 7b. Service advertisement packet is not used in IDEAL and FLOOD, hence it has no effect on them.

Percent of succeeded SDP sessions in the group-based protocols are higher than IDEAL, this is because that service advertisement packets spreads service information in the network. This is to say, the service description of a service may be cached by some other nodes, and these nodes can also respond to service requests, which is the so called server-manifold-effect in Reference [11].

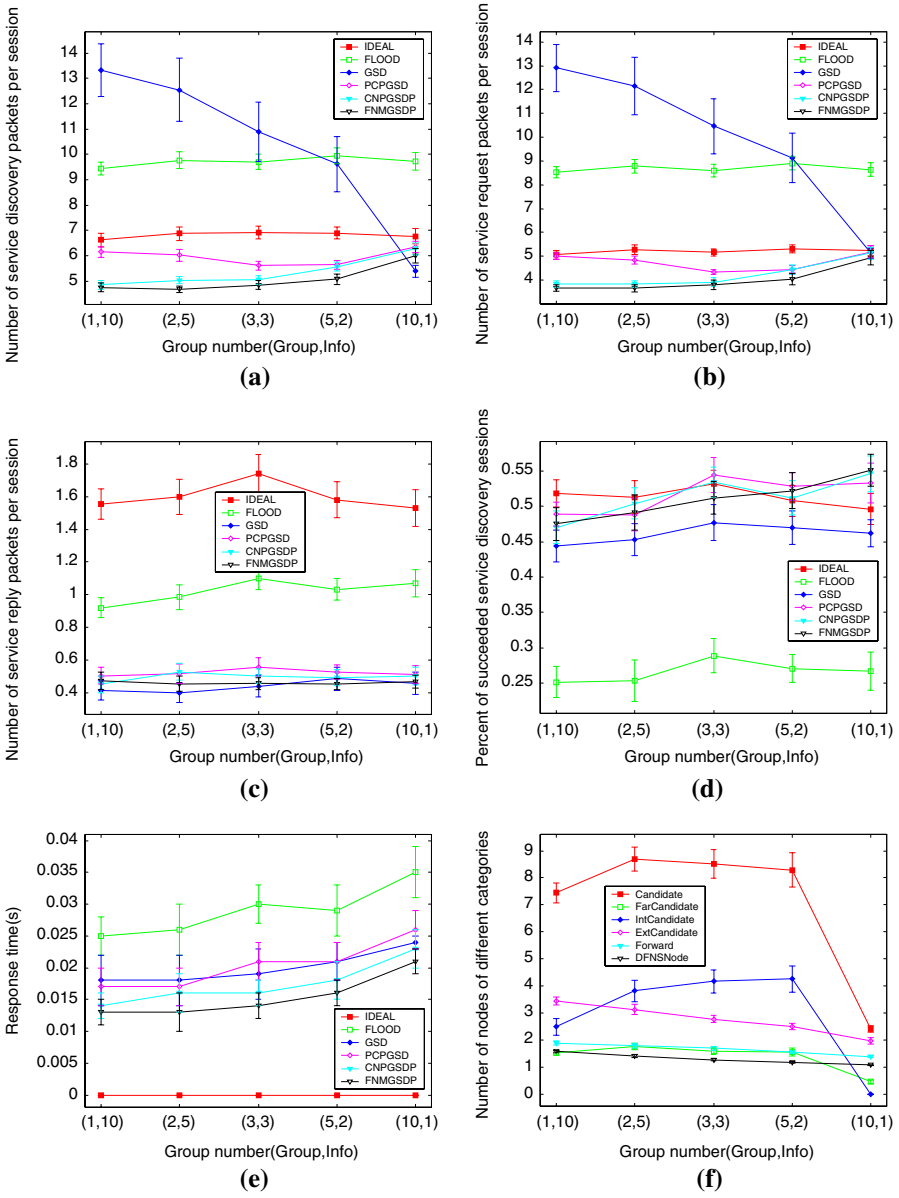


Fig. 8 Effects of (group, info) configuration. **a** number of service discovery packets per session; **b** number of service request packets per session; **c** number of service reply packets per session; **d** percent of succeeded service discovery sessions; **e** response time; **f** number of candidate nodes of different categories

4.4.3 Effects of (Group, Info) Configuration

To inspect the effect of (group, info) configuration on protocol performance, five additional simulation sets are performed. The two numbers in (group, info) represent group number and service number in each group, respectively. The (group, info) configuration in these

simulation experiments are set to (1, 10), (2, 5), (3, 3), (5, 2), and (10, 1), respectively. In all these simulations, radio range is fixed to 100 m. Experiment results are shown in Fig. 8.

As group number increases and service number in each group decreases, the guiding effect of the group information when forwarding service request packets becoming diminished. There will be more cases that there are no candidate nodes. Hence, more request packets will be forwarded in FLOOD mode. As a result, the group-based protocols are degraded to FLOOD schemes, which is confirmed by results in Fig. 8a, b. With the help of server-manifold-effect, the group-based protocols outperform FLOOD. Averaged number of service reply packets in GSD is only about 0.4, which is much smaller than those of other group-based protocols, as shown in Fig. 8c. Hence, the percent of succeeded SDP sessions in GSD is lower than those of other group-based protocols, as shown in Fig. 8d.

In a summary, simulation results show the superiority of FNMGSDP, and the schemes made to FNMGSDP are efficient in reducing service request packets.

5 Conclusions

In this paper, FNMGSDP is proposed to improve CNPGSDP protocol. FNMGSDP minimizes the number of forward nodes when forward request packets by making full use of the information in SIC. Simulation results showed the efficiency of FNMGSDP. Although FNMGSDP requires a little more computation ability, but considering that packet transmission consumes the main part of power consumption of wireless nodes, and power energy is more critical than computing resource and memory resource, FNMGSDP is more preferable than its predecessors.

In the current design, the number of forward nodes of a service request packet is minimized by using a greedy heuristic algorithm. However, if the number of forward nodes is 0, the service request packet will be broadcasted to all neighbors. The spreading of broadcasted service request packets leads to much packet overhead. We are studying new efficient algorithms to resolve this problem. Additionally, nodes that receive reply packets only forward the packets further, not taking full advantage of the service information in these packets. We suspect that service discovery can benefit a lot from the service information in reply packets. We are investigating a new version service discovery protocol that makes use of the service information in reply packets. Study on adaptive service information cache algorithms is also planed.

Acknowledgments This work is jointly supported by: National Science Foundation of China (No. 60703090); Young Promising Researcher Supporting Project of Harbin Engineering University (No. 0811).

References

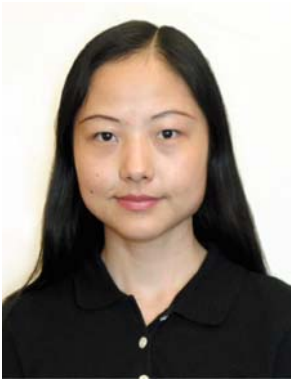
1. IETF. Mobile ad-hoc network (MANET) working group. Mobile ad-hoc networks (MANET). Available: http://www.ietf.org/html.charters/manet-charter.html. 2007.03.
2. Hermann, R., Husemann, D., Moser, M., Nidd, M., Rohner, C., & Schade, A. (2001). DEAPspace–transient ad hoc networking of pervasive devices. *Computer Networks, 35*(4), 411–428.
3. Nidd, M. (2001). Service discovery in DEAPspace. *IEEE Personal Communications, 8*(4), 39–45.
4. Motegi, S., Yoshihara, K., & Horiuchi, H. (2002). Service discovery for wireless ad hoc networks. In *Proceedings of 5th international symposium wireless personal multimedia communications, (WPMC'02)* (pp. 232–236).
5. Engelstad, P. E., & Zheng, Y. (2005). Evaluation of service discovery architectures for mobile ad hoc networks. In *Proceedings of the 2nd annual conference on wireless on-demand networks and services (WONS'05)* (pp. 2–15). St. Moritz, Switzerland.

6. Ververidis, C. N., & Polyzos, G. C. (2005). Routing layer support for service discovery in mobile ad hoc networks. In *Proceedings of the 3rd IEEE international conference on pervasive computing and communications-pervasive wireless networking workshop (PerCom'05)* (pp. 258–262) Kauai Island, Hawaii, USA.
7. Helal, S., Desai, N., Verma, V., & Lee. C. (2003). Konark—a service discovery and delivery protocol for ad-hoc networks. In *Proceedings of the 3rd IEEE conference on wireless communication networks (WCNC'03)* (pp. 2107–2133).
8. Chakraborty, D., Joshi, A., Yesha, Y., & Finin, T. (2002). GSD: A novel group-based service discovery protocol for MANETs. In *Proceedings the 4th IEEE conference on mobile and wireless communications networks (MWCN'02)* (pp. 140–144).
9. Chakraborty, D., Joshi, A., Yesha, Y., & Finin, T. (2006). Towards distributed service discovery in pervasive computing environments. *IEEE Transactions on Mobile Computing, 5*(2), 97–112.
10. Gao, Z.G., Wang, L., Yang, X.Z., & Wen, D.X. (2006). PCPGSD: An enhanced GSD service discovery protocol for MANETs. *Computer Communications, 29*(12), 2433–2445.
11. Gao, Z.G., Wang, L., Yang, M., & Yang, X.Z. (2006). CNPGSDP: An efficient group-based service discovery protocol for MANETs. *Computer Networks, 50*(16), 3165–3182.
12. Ratsimor, O., Chakarborty, D., Joshi, A., & Finin, T. (2002). Allia: Alliance-based service discovery for ad-hoc environments. *Proceedings of the 2nd ACM international workshop on mobile commerce (WMC'02)* (pp. 1–9). Atlanta, Georgia, USA.
13. Kozat, U.C., & Tassiulas, L. (2003). Service discovery in mobile ad hoc networks: An overall perspective on architectural choices and network layer support issues. *Ad Hoc Networks, 2*(1), 23–44.
14. Nordbotten, N.A., Skeie, T., & Aakvaag, N.D. (2004). Methods for service discovery in bluetooth scatternets. *Computer Communications, 27*(11), 1087–1096.
15. Nuevo, J., & Grégoire, J. C. (2004). Proposition of a hierarchical service distribution architecture for ad hoc networks based on the weighted clustering algorithm. *Proceedings of the 5th european wireless conference (EWC'04)*. Barcelona, Spain.
16. Liu, J.C., Zhang, Q., Zhu, W.W., & Li, B. (2003). Service locating for large-scale mobile ad Hoc network. *International Journal of Wireless Information Networks, 10*(1), 33–40.
17. Yoon, H. J., Lee, E. J., Jeong, H., & Kim, J. S. (2004). Proximity-based overlay routing for service discovery in mobile ad hoc networks. *Proceedings of the 19th international symposium on computer and information sciences (ISCIS'04)* (pp. 176–186).
18. Klein, M., Ries, B. K., & Oberiter, P. (2003). Lanes—a lightweight overlay for service discovery in mobile ad hoc networks. In *Proceedings of the 3rd workshop on applications and services in wireless networks (ASWN'03)* (pp. 101–112). Berne, Switzerland.
19. Klein, M., Ries, B. K., & Obreiter, P. (2003). Service rings—a semantic overlay for service discovery in ad hoc networks. In *Proceedings of the 14th international workshop on database and expert systems applications (DEXA'03)* (pp. 180–185). Prague, Czech.
20. Klein, M., & Ries, B. K. (2002). Multi-layer clusters in ad-hoc networks—an approach to service discovery. In *Proceedings of the 1st international workshop on peer-to-peer computing (IWP2PC'02)* (pp. 187–201). Pisa, Italy.
21. Gao, Z. G., Yang, Y. T., Zhao, J., Cui, J. W., & Li, X. (2006). Service discovery protocols for MANETs: A survey. In *Proceedings of the 2nd international conference on mobile ad hoc and sensor networks (MSN'06)* (pp. 232–243), Hong Kong, China.
22. Chvatal, V. (1979). A greedy heuristic for the set-covering problem. *Mathematics of Operations Research, 4*(3), 233–235.
23. Gao, Z. G. (2007). *GloMoSim network simulator—from a beginner to an expert* (pp. 137–157). Harbin: Press of Harbin Institute of Technology.

Author Biographies

Zhenguo Gao He is now a professor in Harbin Engineering University, Harbin, China, received his BS and MS degree in Mechanical and Electrical Engineering from Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively. Then he received his Ph.D degree in Computer Architecture from Harbin Institute of Technology, Harbin, China, in 2006. He is now a faculty member of College of Automation of Harbin Engineering University. His research interests include wireless ad hoc network, sensor network, pervasive computing, etc. He is a senior member of China Computer Federation. He received National Science Foundation Career Award in 2007 and Outstanding Junior Faculty Award of Harbin Engineering University in 2008. He is severing as a viewer for project proposals for National science foundation of China, Ministry of Education of China, Science Foundation of Heilongjiang Province, China. He is also serving as a viewer for many journals including IEEE Transactions on Mobile Computing, Wireless Networks and Mobile Computing, Journal of Parallel and Distributed Computing, IETE Technical Review, Journal of Electronics (Chinese), Journal of Astronautics (Chinese). He was a session chair in ICMA'2007 and organized a Special Session in CCNC'2010.

Ling Wang received her BS degree in mathematics from Heilongjiang University in 1992 and the MS degree in control engineering from Heilongjiang University, in 1995. She received her Ph.D degree in electrical engineering from University of Nevada, Las Vegas, USA, in 2003. In 2004, she joined the faculty of the College of Computer Science and Technology as an assistant professor. Her primary interests are in VLSI design and various aspects of computer-aided design including wireless network, hardware-software co-design, high-level synthesis, and low-power system design.



Mei Yang received her Ph.D. degree in Computer Science from the University of Texas at Dallas in 2003. She was appointed as an Assistant Professor in the Department of Computer Science at Columbus State University from Aug. 2003 to Aug. 2004. She is now an Assistant Professor in the Department of Electrical and Computer Engineering at University of Nevada, Las Vegas. Her research interests include wireless sensor networks, computer architectures, and embedded systems.

Jianping Wang received her Ph.D degree in Computer Science from the University of Texas at Dallas in 2003. She was appointed as an Assistant Professor in the Department of Computer Science at Columbus State University from Aug. 2003 to Aug. 2004. She is now an Associate Professor in the Department of Computer Science, City University of Hong Kong Kowloon, Hong Kong. Her research interests include wireless sensor networks, service-oriented computing.