Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 50 (2006) 3165–3182
Computer Networks
www.elsevier.com/locate/comnet

CNPGSDP: An efficient group-based service discovery protocol for MANETs

Zhenguo Gao [a,*], Ling Wang [a], Mei Yang [b], Xiaozong Yang [a]

[a] Harbin Institute of Technology, Department of Computer Science and Technology, Harbin 150001, China
[b] University of Nevada, Department of Electrical and Computer Engineering, Las Vegas, NV 89119, United States

Received 27 June 2005; received in revised form 2 December 2005; accepted 14 December 2005
Available online 13 January 2006

Responsible Editor: F. Cuomo

Abstract

The ability to discover services is the major prerequisite for effective usability of MANETs. Group-based Service Discovery (GSD) protocol is a typical service discovery protocol for MANETs. However, because of large redundant packet transmissions, its packet overhead is high. In this paper, in light of GSD, we propose a new service discovery protocol for Mobile Ad-Hoc Networks (MANETs): Candidate Node Pruning enhanced Group-based Service Discovery Protocol (CNPGSDP). In CNPGSDP, two schemes are introduced to enhance GSD: Broadcast Simulated Unicast (BSU) and Candidate Node Pruning (CNP). In BSU, several unicast request packets are replaced with one request packet transmitted in broadcast mode with all unicast receivers enclosed. CNP further reduces the number of request packets by reducing the number of candidate nodes. Mathematical analysis and simulation tests both show that CNPGSDP is a very effective, efficient, and prompt service discovery protocol for MANETs.

Keywords: GSD; Mobile Ad-Hoc Networks; Service discovery protocol

1. Introduction

Mobile Ad-Hoc Networks (MANETs) [1] are temporary infrastructure-less multi-hop wireless networks that consist of many autonomous wireless mobile nodes. Flexibility and minimum user intervention are essential for such future communication networks [2]. Service discovery, which allows devices to advertise their own services to the rest of the network and to automatically locate network services with requested attributes, is a major component of MANETs.

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 its types and attributes, access method,

* Corresponding author. Tel.: +86 451 86413754; fax: +86 451 86414093.
E-mail addresses: gag@ftcl.hit.edu.cn (Z. Gao), lwang@ftcl.hit.edu.cn (L. Wang), meiyang@egr.unlv.edu (M. Yang), yxz@hit.edu.cn (X. Yang).

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2005.12.004

etc.; A server is a node that provides some services; A client is a node that requests services provided by other nodes. When a node needs services from others, it generates a service request packet. When receiving the request packet, each node that provides matched services responds with a service reply packet. Nodes without matched services forward the packet further. All these packet transmissions, including request packets and reply packets, form a Service Discovery Protocol (SDP) session.

The objective of service discovery protocol is to reduce service request packet redundancy while retaining service discoverability. Service discovery has been originally studied in the context of wired networks. Several different industrial consortiums and organizations have been established to standardize various service discovery protocols, such as IETF's Service Location Protocol (SLP) [3], Sun's Jini [4], Microsoft's Universal Plug and Play (UPnP) [5], IBM's Salutation [6], Object Management Group (OMG)'s Common Object Request Broker Architecture (CORBA) [7], etc.

With the development of wireless technology, several service discovery protocols and technologies used in wireless context have been proposed [8–10]. However, these protocols are designed for one-hop wireless networks only. They are not applicable to multi-hop MANETs.

Many recent researches focus on service discovery protocols for MANETs [11–22]. Some of them are adapted from service discovery protocols in wired networks [11–13], but they are not very suitable for MANETs for the restriction of the basal protocol architecture. Other efforts are targeted at MANETs 0–0. According to the methods used to reduce packet redundancy, service discovery protocols targeted at MANETs can be classified into two classes: probability-based schemes [14–18], and semantic-routing-based schemes [19–21].

1.1. Probability-based service discovery schemes

In probability-based service discovery schemes [14–18], when receiving a request packet, each node that does not know about any matched services will forward the packet with probability P.

When P is fixed as 1, the scheme degenerates to flooding. Flooding is used in secure service discovery [14], Konark [16], and [15], etc. However, flooding may lead to great packet redundancy, serious contention, and frequent collisions (called as the broadcast storm problem) [23].

When P is less than 1, the redundancy of request packets will be reduced. Meanwhile the coverage of request packets will also be reduced as well in common MANETs, which greatly affects the ability of finding matched services. Additionally, when P is constant and the maximum number of hops that request packet can travel is relatively small (this is true in service discovery tasks), the number of succeeded requests will decrease almost in the same speed as that of the value of P. This drawback has been proved through simulations in [17,18] and [24]. Thus, in Flexible Forward Probability based Service Discovery Protocol (FFPSDP) [17], probability P is made to decrease gradually along with the travel of request packets. Reply Info Cache enhanced FFPSDP (RICFFP) [18] enhanced FFPSDP by caching the service information in reply packets temporarily. This cached information can be used as valid source of matched services.

1.2. Semantic-routing-based service discovery schemes

In semantic-routing-based schemes [19–21], nodes can intelligently select next-hop nodes for request packets by inspecting service description semantics as well as local topology. When receiving the request packet, these selected nodes will forward the packet, while other nodes will not. Such schemes are used in Group based Service Discovery protocol (GSD) [20], Service Ring [21], etc.

In Service Ring, nodes are organized into multi-layer hierarchical rings. This architecture has good scalability. But maintaining such a complicated architecture in highly dynamic MANETs is hard and costly, which has been proved through simulations [22]. Hence, hierarchical Server Ring is not very suitable for MANETs.

In GSD, services are classified into several groups. Each server generates service advertisement packets periodically. A service advertisement packet includes the information about services provided by the sender and the groups that the services provided by some servers in the sender's vicinity belong to. When forwarding request packets, some neighbors of the current node may have seen some services belonging to the same group as requested service. Such nodes are referred as candidate nodes. Service requests will be matched at those nodes with higher probability. Thus, instead of broadcasting the request packet to all neighbours, GSD selectively

forwards the request packet towards these candidate nodes in unicast mode.

Klein et al. [21] argue that semantic routing is a step in the right direction towards service discovery. Since that the hierarchical architecture of Server Ring is hard to maintain, GSD is more acceptable. However, for each candidate node, GSD forwards a request packet in unicast mode, which results in serious redundancy. Hence, in this paper, we propose a new group-based service discovery protocol: Candidate Node Pruning enhanced Group-based Service Discovery Protocol (CNPGSDP). In CNPGSDP, several unicast request packets are replaced with one request packet sent in broadcast mode. Additionally, with the help of a little additional information, the number of valid receivers of a request packet sent in broadcast mode is significantly reduced. Consequently, the number of successive request packets sent by the receivers is reduced.

The rest of the paper is organized as follows. In Section 2, an overview of GSD is given. In Section 3, CNPGSDP is presented. In Section 4, mathematical analysis for CNPGSDP is presented to show the number of request packets that can be saved. In Section 5, comparative studies on CNPGSDP and several other service discovery protocols are performed through extensive simulations. Simulation results are shown and explained in detail. Finally, in Section 6, a conclusion is presented.

2. Overview of GSD

Three basic operations in GSD are service advertisement packet spreading, service request packet forwarding, and service reply packet routing. Two effective mechanisms, peer-to-peer caching of service advertisement packets and group-based intelligent forwarding of service request packets, are used in the first two operations, respectively. Benefiting from these mechanisms, GSD achieves efficient network bandwidth usage and increased flexibility in the service matching process.

2.1. Service advertisement packet spreading

Each server will generate service advertisement packets periodically. These packets can be forwarded further. To restrict the spreading range of service advertisement packets, the maximum number of hops they can travel is limited (denoted as d). The content of a service advertisement packet contains not only the description of the service provided by the server, but also the groups that the services provided by the node's in the server's d-hop neighbor set belong to. A node's d-hop neighbor set is the set of nodes that are at most d hop away from node u.

Each node maintains a cache called Service Information Cache (SIC), which is used to store service advertisement packet temporally. This is the so-called peer-to-peer caching of service advertisement packets. By caching service advertisement packets, a node knows not only the services provided by the servers in its d-hop neighbor set, but also the groups that the services provided by these server's d-hop neighbors belong to.

Fig. 1 shows an example of service advertisement packet spreading with $d = 1$. Symbols in this figure are listed in Table 1.

In Fig. 1, after several cycles of service advertisement spreading operation, each node has constructed its own SIC. Hence, when new operation cycle comes, each node should construct new service advertisement packet basing on its SIC (the groups of services cached in SIC are also enclosed in new packet). New service advertisement packets of all servers are shown in the figure. Before sending a packet, the number of hops that the packet can travel is decreased by 1. Hence, although the hop limit of the service advertisement packets is 1, the remaining hop of these packets is 0.

2.2. Service request packet forwarding

When a node needs services and there is no matched services in its SIC, the node constructs a service request packet and forwards the packet

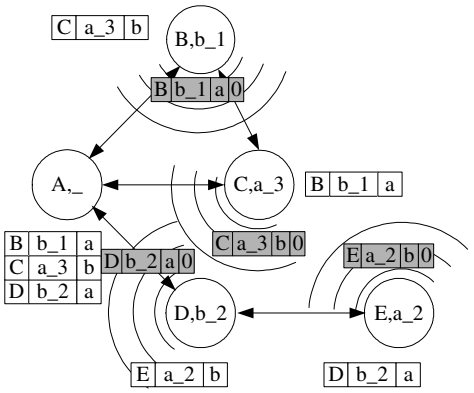


Fig. 1. Example of service advertisement packet spreading in GSD.

Table 1
Tokens used in demonstrating MANETs

Symbol	Indication	Example
Circle	Mobile node	
String in a circle	The identity of the node and the services it provides	The circle with string "*B*, *b*_1" in Fig. 1 indicates that: (1) the node is *B*; (2) node *B* provides a service "*b*_1"; which belongs to service group "*b*"
White table adjacent to a node	The Service Information Cache (SIC) of the node (not all fields are shown)	The first entry of *A*'s SIC {*B*,"*b*_1","*a*"} in Fig. 1 indicates: (1) the server corresponds to the entry is node *B*; (2) node *B* provides service "*b*_1", (3) some nodes in the *d*-hop neighbor set of node *B* provide group "*a*" services
Double-headed arrow	Two nodes on both ends are neighbors	In Fig. 1, Nodes *C* and *A* are mutual neighbors, while nodes *C* and *E* are not neighboring
Arcs around a node	Indicate packet transmission	In Fig. 1, node *B* sends out a packet, while nodes *A* not
Grey table over arcs	The content of the packet being transmitted (not all fields are shown)	In Fig. 1, grey tables represent the content of service advertisement packet. The packet {*B*,"*b*_1","*a*",0} sent by node *B* indicates that: (1) the sender is *B*; (2) node *B* provides a service "*b*_1"; which belongs to service group "*b*"; (3) some nodes in the *d*-hop neighbor set of node *B* provide group "*a*" services; (4) the packet can still travel 0 hops

towards some elaborately selected nodes in unicast mode. These nodes are selected with the criterion that some nodes in its *d*-hop neighbor set provide some services belonging to the same group as the requested service. The packet will get matched at these selected nodes with high probability. When receiving the packet sent by the current node, each selected node should forward the packet further, unless the packet is matched or exceeds its hop limit. An example of request packet routing in GSD is shown in Fig. 2. In this figure, most symbols have the same meaning as in Fig. 1 except for the arc and the grey table. In Fig. 2, arcs around a node indicate service request packet transmissions. Correspondingly, the grey table over the arcs represents the content of the service request packet being transmitted (not all fields are shown).

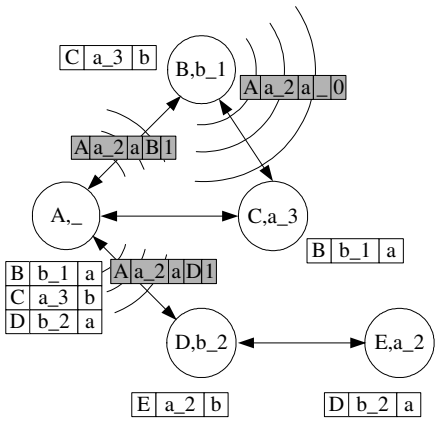


Fig. 2. Example of service request packet forwarding in GSD.

When node *A* needs service "*a*_2" belonging to group "*a*", and there is no matched service in its SIC, it has to select some nodes based on its SIC. Node *A* finds that nodes *B* and *D* both have some servers in its *d*-hop neighbour set providing services belonging to group "*a*". Thus, nodes *B* and *D* are both selected and two unicast request packets are sent to them, respectively. For example, the request packet {*A*,"*a*_2","*a*",*B*,1} sent to *B* indicates that: (1) the source that generates the request packet is node *A*; (2) the requested service is "*a*_2"; (3) the requested service belongs to group "*a*"; (4) the packet is sent to node *B*; (5) the request packet can still travel 1 hop. Thus, when receiving the packet, node *B* has to forward it further. Node *D*'s SIC shows that node *E* provides "*a*_2". Hence, when node *D* receivers the packet from node *A*, it responds with a reply packet in stead of forwarding the request packet further.

2.3. Service reply packet routing

If the node that receives a new request packet finds matched services, it sends out a service reply packet in unicast mode to the direct sender of the service request packet. The service reply packet will be relayed to the source of the service request packet along the reverse path.

2.4. Suggested improvements

Service advertisement packets do not enclose detailed service descriptions. They store simple service group information instead. Hence, there is no significant increase in advertisement packet size. Therefore, it does not impose much stress on wireless bandwidth. Meanwhile, benefiting from service group information, request packets are intelligently forwarded to relay nodes. Hence, the number of request packets is reduced. However, there is still much room to reduce the number of request packets. The reasons are given as following:

- There will be too many unicast request packets. It may be very often that many candidate nodes are found, such as shown in Fig. 2. In such cases, many unicast request packets will be sent resulting in large redundancy of request packets.
- The number of candidate nodes can be reduced. As shown in Fig. 2, node *A* knows that node *C* provides service "*a*_3" only. If node *A* also know that the service belonging to group "*a*" seen by node *B* is provided by node *C*, *A* can determine that *B* must not know about any matched service. Thus, in this case, node *B* can be removed from candidate node set, and the request packet sent towards node *B* can be saved.

Hence, considering above situations, we propose the CNPGSDP based on the idea of group-based intelligent forwarding of request packets in GSD.

3. Candidate Node Pruning enhanced Group-based Service Discovery Protocol (CNPGSDP)

3.1. New schemes in CNPGSDP

Two schemes are proposed in CNPGSDP to enhance GSD: Candidate Node Pruning (CNP) and Broadcast Simulated Unicast (BSU).

3.1.1. Candidate node pruning (CNP) scheme

Different from GSD where all candidate nodes are retained, in CNPGSDP, some special candidate nodes so called as internal candidate nodes are pruned. This scheme is named as Candidate Node Pruning (CNP).

In order to implement CNP scheme, some additional information facilitating the decision of candidate node pruning should be collected through packet exchange. Hence, modifications are made to data structures. By using the CNP scheme, successive request packets sent by next-hop nodes are reduced.

3.1.2. Broadcast Simulated Unicast (BSU) scheme

In CNPGSDP, instead of sending one unicast request packet towards each candidate node as in GSD, only one request packet piggybacked with the list of so called relay nodes of the candidate nodes is transmitted in broadcast mode. This scheme is named as Broadcast Simulated Unicast (BSU).

In order to implement BSU scheme, a new compound field called *receiver-list* is inserted in the request packet. The *receiver-list* field stores the list of receivers that will forward the request packet. By using the BSU scheme, many unicast request packets can be saved.

3.2. Data structures in CNPGSDP

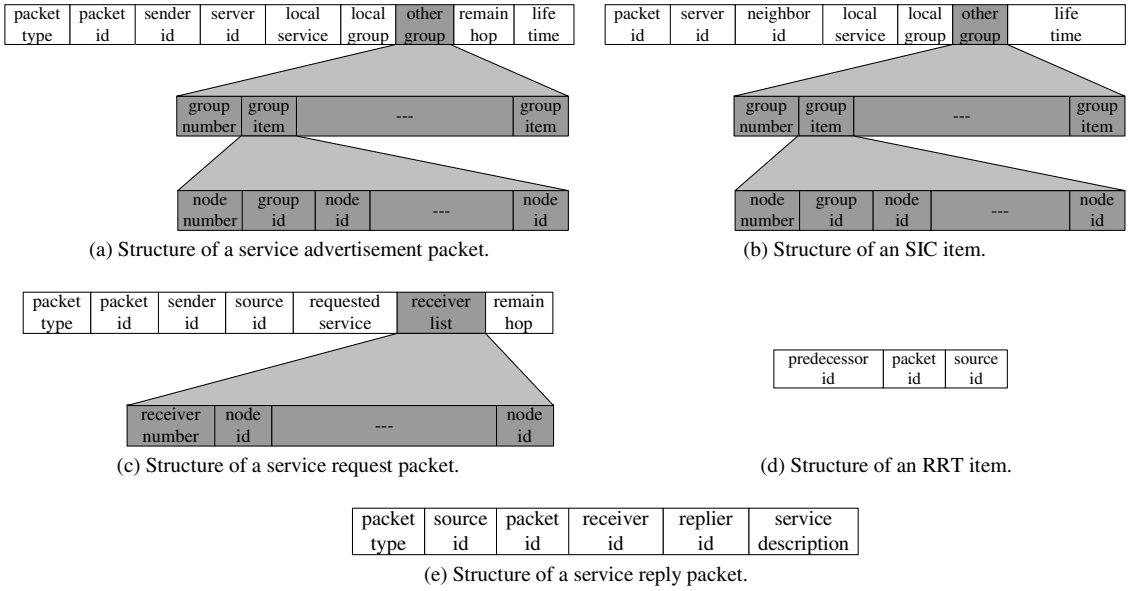
Compared with GSD, slight modifications are made to some data structures. Main data structures in CNPGSDP are shown in Fig. 3. Modified or new fields are highlighted with grayed background.

3.2.1. Structure of service advertisement packet

The structure of the service advertisement packet in CNPGSDP is shown in Fig. 3(a). Only the *other-group* field is changed. In GSD, this field encloses the list of service groups that the services provided by nodes in the *d*-hop neighbor set of the server belong to. Whereas in CNPGSDP, for each group indicated by *group-id*, the list of servers is also included in the *group-item* compound field.

Other fields unchanged are listed as follows:

packet-type indicates the packet type;
packet-id a number increases monotonically with each service advertisement packet generated by the node. This field 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 these services belong to;



(a) Structure of a service advertisement packet.

(b) Structure of an SIC item.

(c) Structure of a service request packet.

(d) Structure of an RRT item.

(e) Structure of a service reply packet.

Fig. 3. Data structures in CNPGSDP.

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 the node's SIC.

3.2.2. Structure of SIC

SIC is used to cache service advertisement packets. Hence, similar modifications are made to the structure of SIC entry, as shown in Fig. 3(b). All fields are the same as those of the service advertisement packet except for *neighbor-id* field which indicates the node from which the service advertisement packet is received.

3.2.3. Structure of service request packet

The structure of the service request packet is shown in Fig. 3(c). Compared with GSD, a new filed *receiver-list* is inserted. The *receiver-list* compound field stores the list of receivers selected by the sender. The *receiver-number* subfield of indicates the number of receivers in the list. If the *receiver-number* field is 0, then every node that receives the packet is a valid receiver. Other fields unchanged are listed as follows:

packet-type indicates the packet type;

packet-id a number increasing monotonically with each request packet from a node;

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;

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.

3.2.4. Structure of RRT

Each node maintains a RRT used in two tasks: (1) check duplicated request packets, and (2) route service reply packets to the corresponding source node. Fig. 3(d) shows the structure of an RRT entry, which is the same as that of GSD. The *predecessor-id* field indicates the node from which the request packet is received. The node indicated by this field is just the next hop node that a corresponding reply packet will be forwarded to. The *packet-id* field and the *source-id* field are as same as those of a request packet.

3.2.5. Structure of service reply packet

Compared with those in GSD, the structure of the service reply packet in CNPGSDP remains

unchanged, as shown in Fig. 3(e). They are described as follows:

packet-type indicates the 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 stores the description of the matched services.

3.3. Operations of CNPGSDP

Service reply packet routing in CNPGSDP is as same as that in GSD. Therefore, we will only explain service advertisement packet spreading and service request forwarding in the following sections.

3.3.1. Notations

The following notations are used in the following discussion.

$\boldsymbol{d}$	the maximum number of hops that advertisement packets can travel;
$\boldsymbol{g}$	the group that the requested service belongs to;
$\boldsymbol{u}$	the current node;
$\boldsymbol{N_x(w)}$	the set of nodes that are at most x-hop away from node w, i.e., node w's x-hop neighbor set (excluding none w itself);
$\boldsymbol{S(w)}$	the set of servers that have corresponding valid SIC entries in node w's SIC. Note that each server in $N_d(w)$ has a corresponding entry in node w's SIC;
$\boldsymbol{e(w,s)}$	the entry that corresponds to server s in node w's SIC;
$\boldsymbol{E(w)}$	the set of entries in node w's SIC;
$\boldsymbol{r(w,s)}$	the node indicated by the *neighbor-id* field of $e(w,s)$. That is, $r(w,s) = e(w,s).neighbor\text{-}id$;
$\boldsymbol{G(w,s,g)}$	the set of nodes in the *group-item* field in $e(w,s)$'s *other-group* field whose *group-id* field equals to g;
$\boldsymbol{S(w,g)}$	$S(w,g) = \{s \mid s \in S(w), G(w,s,g) \neq \varnothing\}$;
$\boldsymbol{S_I(w,g)}$	$S_I(w,g) = \{s \mid s \in S(w,g), G(w,s,g) \subseteq S(w) \cup \{w\}\}$;
$\boldsymbol{S_E(w,g)}$	$S_E(w,g) = S(w,g) - S_I(w,g)$.

3.3.2. Definitions

- **Definition 1:** *Candidate node*. Nodes in $S(u,g)$ are all candidate nodes of node u.
- **Definition 2:** *Relay node*. $r(u,s)$ is the relay node of candidate node s. In other words, the relay node of a candidate node is the next-hop node on the path from the current node u to its candidate node s.
- **Definition 3:** *Internal candidate node*. Candidate nodes in $S_I(u,g)$ are called as internal candidate nodes.
- **Definition 4:** *External candidate node*. Candidate nodes in $S_E(u,g)$ are called as external candidate nodes.

3.3.3. Service advertisement packet spreading

The process of service advertisement packet spreading in CNPGSDP is the same as that in GSD. The difference only exists in the structures of service advertisement packet and the SIC entry. Fig. 4 shows an example with service advertisement packet restricted to 1 hop.

In Fig. 4, after several cycles of service advertisement spreading operation, each node has constructed its own SIC. When new operation cycle comes, each node constructs new service advertisement packet basing on its SIC. In CNPGSDP, new advertisement packet enclose not only the list of service groups that the server has seen in its d-hop neighbor set, but also the list of servers for each service group. New service advertisement packets of all servers are shown in the figure.

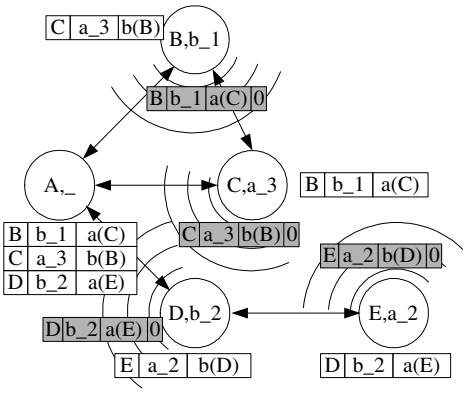


Fig. 4. Example of service advertisement packet spreading in CNPGSDP.

3.3.4. Service request packet forwarding

When receiving an unduplicated service request packet, a node that does not know any matched service will forward the 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.

The current node performs the following four steps in sequence to forward the service request packet:

Step 1. Determine candidate node set $S_{\mathrm{CNPGSDP}}(u,g)$

$$S_{\mathrm{CNPGSDP}}(u,g) = S(u,g) - S_{\mathrm{I}}(u,g).$$

Step 2. Determine Relay Node Set $R_{\mathrm{CNPGSDP}}(u,g)$

$$R_{\mathrm{CNPGSDP}}(u,g) = \{r(u,s) | s \in S_{\mathrm{CNPGSDP}}(u,g)\}.$$

Step 3. Enclose the List of Relay Nodes Enclose nodes in $R_{\mathrm{CNPGSDP}}(u,g)$ in the service request packet's receiver-list compound field and set the *receiver-number* field of the *receiver-list* field to the number of nodes in $R_{\mathrm{CNPGSDP}}(u,g)$.

Step 4. Send the request packet in broadcast mode.

The scheme of pruning $S_{\mathrm{I}}(u,g)$ in step 1 is the so called CNP scheme, and the operations in step 3 and step 4 is the so called BSU scheme.

The correctness of pruning $S_{\mathrm{I}}(u,g)$ is guaranteed by the following Theorem 1.

Theorem 1. *When forwarding a request packet, all internal candidate nodes in $S_I(u,g)$ can be removed from $S(u,g)$.*

Proof. We denote the current node as u. Since that node u will forward a request packet, node u must not know about any matched service. That is, neither node u itself nor its SIC has any matched service information. Based on this fact, we now prove that all internal candidate nodes in $S_{\mathrm{I}}(u,g)$ do not know about any matched service, and consequently, they can be removed from $S(u,g)$. We will prove it by contradiction.

Suppose that there is a node $w \in S_{\mathrm{I}}(u,g)$ and w knows about a matched service. There are two cases.

- If the matched service is provided by node w itself, then the matched service must be in the local service field of $e(u,w)$ entry in node u's SIC. This is contrary to the fact that node u must not know about any matched service.
- If the matched service is provided by some servers in $N_d(w)$, the set of these servers is $G(u,w,g)$. According to the definition of $S_{\mathrm{I}}(u,g)$, there is $G(u,w,g) \subseteq S(u,g) \cup \{u\}$. Thus there are two sub-cases to be considered:
  - If node $u \in G(u,w,g)$, this is contrary to the fact that node u must not know about any matched service.
  - If $G(u,w,g) \subseteq S(u,g)$, there must be corresponding entries in node u's SIC. Thus, the matched service must be in these entries' *local-service* field. This is also contrary to the fact that node u must not know about any matched service. □

CNP scheme reduce the number of successive request packets sent by next hop nodes by reduce the number of candidate nodes. BSU scheme can reduce the number of request packets further by replaces several unicast packets with one packet sent in broadcast mode.

3.4. Example of request packet forwarding in CNPGSDP

Fig. 5 shows an example of forwarding a request packet in CNPGSDP. Suppose the value of the *remain-hop* field of the request packet is greater than 1. When node A needs service "a_2", which belongs to service group "a", it checks its SIC first. Obviously, node A finds no matched service. Then, node A begins to select candidate nodes and relay nodes based on its SIC.

Node A knows:

$$S(A) = \{B, C, D\},$$
$$G(A,B,g) = \{C\},$$
$$G(A,D,g) = \{E\},$$
$$r(A,B) = B,$$
$$r(A,D) = D.$$

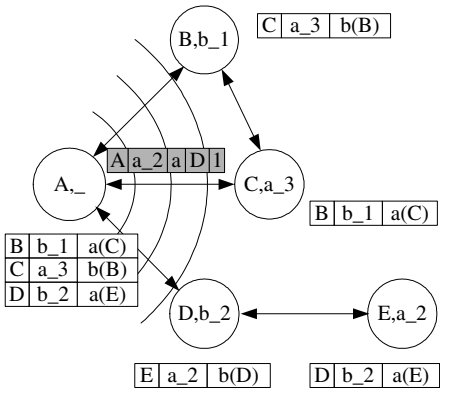


Fig. 5. Example of request packet forwarding in CNPGSDP.

Hence,

$$S(A,g) = \{B,D\},$$
$$G(A,B,g) \subseteq S(A),$$
$$G(A,D,g) \not\subseteq S(A),$$
$$S_{\mathrm{I}}(A,g) = \{B\},$$
$$S_{\mathrm{CNPGSDP}}(A,g) = S(A,g) - S_{\mathrm{I}}(A,g) = \{B,D\} - \{B\} = \{D\}.$$

At last, the set of relay nodes in CNPGSDP is:

$$R_{\mathrm{CNPGSDP}}(A,g) = \{r(A,s) | s \in S_{\mathrm{CNPGSDP}}(A,g)\} = \{r(A,D)\} = \{D\}.$$

In CNPGSDP, only node D is selected as the candidate node. Hence, only one request packet with node D as the receiver will be sent in broadcast mode. Node D's SIC shows that node E provides "*a_2*". Hence, when node D receivers the packet from node A, it responds with a reply packet in stead of forwarding the request packet further. Compared with Fig. 2, two service request packets are saved.

4. Theoretical analysis and verification

In this section, mathematical analysis is performed to estimate the number of request packets saved of CNPGSDP over GSD. The theoretical result is verified through simulation in matlab.

4.1. Theoretical analysis

Table 2 describes the notation that will be used in this section.

The analysis in the section is based on the following assumptions:

Table 2
Notations used in mathematical analysis

Notation	Description
P_s	The probability that a node has services
I_s	The number of services in each group
G_s	The number of service groups
d	Maximum number of hops that service advertisement packets can travel
n_d	The average number of nodes that are at most d-hop away from a node (excluding the node itself)

Assumption 1. The initial phase of the network has finished, i.e., the convergence of the content of nodes' SIC has finished.

Assumption 2. Nodes are uniformly distributed in network area.

Assumption 3. Packet transmissions are error free.

Assumption 4. Border effect of the network is not considered.

Theorem 2. *The probability that a node does not find any matched services and it finds k $(0 \leqslant k \leqslant n_d)$ candidate nodes is given by*

$$P_{k,\mathrm{all}} = C_{n_d}^{k} \cdot \frac{(I_s-1)\cdot(1-P_s)^{n_d-k}\cdot P_s^{k+1}\cdot(G_s\cdot I_s-1)^k}{(G_s\cdot I_s)^{k+1}} + \sum_{m=k}^{n_d}\left(C_{n_d}^{m}\cdot C_m^k \cdot \frac{P_s^m\cdot(G_s\cdot I_s-1)^m\cdot(1-P_s)^{n_d-m}\cdot(G_s-P_s)^{n_d\cdot(m-k)+1}\cdot\left(G_s^{n_d}-(G_s-P_s)^{n_d}\right)^k}{G_s^{n_d\cdot(m+1)+1}\cdot I_s^m}\right). \quad (1)$$

Proof. The probability that a node does not find any matched services and it finds $k(k \geqslant 0)$ candidate nodes, $P_{k,\mathrm{all}}$, can be calculated as follows:

$$P_{k,\mathrm{all}} = P_{k,\mathrm{nm}} + P_{k,\mathrm{ns}}, \quad (2)$$

where $P_{k,nm}$ is the probability that the current node has at least one unmatched service that belongs to the same group as the requested service, and it finds k candidate nodes; $P_{k,ns}$ is the probability that the current node has no service that belongs to the same service group as the requested service, and it finds k candidate nodes.

We will now calculate $P_{k,nm}$ and $P_{k,ns}$ in sequence.

- The calculation of $P_{k,nm}$

$P_{k,nm}$ can be calculated as follows:

$$P_{k,\mathrm{nm}} = P_{k,\mathrm{n}} \cdot P_{k,\mathrm{k}}, \quad (3)$$

where $P_{k,n}$ is the probability that the current node provides some unmatched services that belong to the same group as the requested service. $P_{k,k}$ is the probability that, in the current node's d-hop neighbor set, there are just k servers and they provide no unmatched services.

Please notice that all these k servers will be candidate nodes. The explanation is as follows. Since the current node must be in these k servers' d-hop neighbor set, they must be able to receive the current node's service advertisement packets, which enclose the information of its services. Therefore, the group information of the current node' services must have been enclosed in the *other-group* field of all these k servers' service advertisement packets. Thus, the current node must be able to receive these service advertisement packets and cache them. Hence, all these k servers will be found as valid candidate nodes by the current node.

Now we calculate $P_{k,n}$ first. There are G_s groups, and there are I_s services in each group. Hence, there are $G_s \cdot I_s$ services in total. Since that the service provided by the current node does not match but belongs to the same group as the requested service, this service must be among $I_s - 1$ services. Thus, we have

$$P_{k,\mathrm{n}} = P_s \cdot \frac{I_s - 1}{G_s \cdot I_s}. \tag{4}$$

$P_{k,k}$ can be represented as

$$P_{k,\mathrm{k}} = C_{n_d}^{k} \cdot P_{k,\mathrm{k},1} \cdot P_{k,\mathrm{k},2}, \tag{5}$$

where $P_{k,k,1}$ is the probability that k nodes in the current node's d-hop neighbor set provide unmatched services; $P_{k,k,2}$ is the probability that all other $n_d - k$ nodes provide do not provide any service.

Since the probability that one node provides unmatched service is $P_s \cdot (G_s \cdot I_s - 1)/(G_s \cdot I_s)$, the probability of k nodes all provide unmatched services is

$$P_{k,\mathrm{k},1} = \left(P_s \cdot \frac{G_s \cdot I_s - 1}{G_s \cdot I_s}\right)^{k}. \tag{6}$$

Obviously,

$$P_{k,\mathrm{k},2} = (1 - P_s)^{n_d - k}. \tag{7}$$

Substituting $P_{k,k,1}$ and $P_{k,k,2}$ in Eq. (5) with Eqs. (6) and (7), respectively, we get

$$\begin{aligned} P_{k,\mathrm{k}} &= C_{n_d}^{k} \cdot P_{k,\mathrm{k},1} \cdot P_{k,\mathrm{k},2} \\ &= C_{n_d}^{k} \cdot \left(P_s \cdot \frac{G_s \cdot I_s - 1}{G_s \cdot I_s}\right)^{k} \cdot (1 - P_s)^{n_d - k}. \end{aligned} \tag{8}$$

Substituting $P_{k,n}$ and $P_{k,k}$ in Eq. (3) with Eqs. (4) and (8), respectively,

$$\begin{aligned} P_{k,\mathrm{nm}} = P_{k,\mathrm{n}} \cdot P_{k,\mathrm{k}} = C_{n_d}^{k} \\ \cdot \frac{(I_s - 1) \cdot (1 - P_s)^{n_d - k} \cdot P_s^{k+1} \cdot (G_s \cdot I_s - 1)^{k}}{(G_s \cdot I_s)^{k+1}}. \end{aligned} \tag{9}$$

- The calculation of $P_{k,ns}$

$P_{k,ns}$ can be calculated as follows:

$$P_{k,\mathrm{ns}} = P_{k,\mathrm{nos}} \cdot \sum_{m=k}^{n_d} P_{k,\mathrm{mk}}, \tag{10}$$

where $P_{k,nos}$ is the probability that the current node has no service that belongs to the same group as requested service. $P_{k,mk}$ is the probability that (1) there are m servers with unmatched services in the current node's d-hop neighbor set (the probability of this case is denoted as $P_{k,1}$), and (2) among these m servers, there are k special nodes: Each of these special nodes has servers in their d-hop neighbor set that provide services belonging to the same group as the requested service. The probability of this case is denoted as $P_{k,2}$. When forwarding request packets, all these k nodes are candidate nodes. Obviously,

$$P_{k,\mathrm{nos}} = 1 - \frac{P_s}{G_s}, \tag{11}$$

$$P_{k,\mathrm{mk}} = P_{k,1} P_{k,2}. \tag{12}$$

Similar to the calculation of $P_{k,k}$ in Eq. (5), the value of $P_{k,1}$ and $P_{k,2}$ can be calculated as:

$$P_{k,1} = C_{n_d}^{m} \cdot \left(P_s \cdot \frac{G_s \cdot I_s - 1}{G_s \cdot I_s}\right)^{m} \cdot (1 - P_S)^{n_d - m}, \tag{13}$$

$$P_{k,2} = C_{m}^{k} \cdot \left(1 - \left(1 - P_s \cdot \frac{1}{G_s}\right)^{n_d}\right)^{k} \cdot \left(\left(1 - P_s \cdot \frac{1}{G_s}\right)^{n_d}\right)^{m-k}. \tag{14}$$

Based on Eqs. (10)–(14), We have

$$\begin{aligned} P_{k,\mathrm{ns}} &= P_{k,\mathrm{nos}} \cdot \sum_{m=k}^{n_d} P_{k,\mathrm{mk}} = P_{k,\mathrm{nos}} \cdot \sum_{m=k}^{n_d} (P_{k,1} \cdot P_{k,2}) \\ &= \left(1 - \frac{P_s}{G_s}\right) \cdot \sum_{m=k}^{n_d} C_{n_d}^{m} \cdot \left(P_s \cdot \frac{G_s \cdot I_s - 1}{G_s \cdot I_s}\right)^{m} \cdot (1 - P_s)^{n_d - m} \cdot C_m^{k} \\ &\quad \cdot \left(1 - \left(1 - \frac{P_s}{G_s}\right)^{n_d}\right)^{k} \cdot \left(1 - \frac{P_s}{G_s}\right)^{n_d \cdot (m-k)} = \sum_{m=k}^{n_d} \left(C_{n_d}^{m} \cdot C_m^{k} \right. \\ &\quad \left. \cdot \frac{P_s^{m} \cdot (G_s \cdot I_s - 1)^{m} \cdot (1 - P_s)^{n_d - m} \cdot (G_s - P_s)^{n_d \cdot (m-k)+1} \cdot \left(G_s^{n_d} - (G_s - P_s)^{n_d}\right)^{k}}{G_s^{n_d \cdot (m+1)+1} \cdot I_s^{m}}\right). \end{aligned} \tag{15}$$

Substituting $P_{k,nm}$ and $P_{k,ns}$ in Eq. (2) with Eqs. (9) and (15), respectively, the mathematical expression of $P_{k,\text{all}}$, as shown in the theorem, is obtained. □

Theorem 3. *In GSD, when forwarding a service request packet, the probability that a node should send out $k(0 \leqslant k \leqslant n_d)$ unicast request packets, $P_{k,forward}$, is given by*:

$$P_{k,\text{forward}} = \frac{P_{k,\text{all}}}{\left(1 - P_s \cdot \frac{1}{G_s \cdot I_s}\right)^{n_d+1}}. \tag{16}$$

Proof. In GSD, if a node does not find any matched services, it should forward the request packet further. In this case, if it finds k candidate nodes, the node will have to send out k unicast request packets to each candidate node. Hence, $P_{k,\text{forward}}$ is just the conditional probability of finding $k(0 \leqslant k \leqslant n_d)$ candidate nodes on condition that it does not find any matched services.

$$P_{k,\text{forward}} = P_{k,\text{all}}/P_{\text{nomatch}}. \tag{17}$$

Each node knows about not only the services provided by the node itself, but also the services provided by nodes in its d-hop neighbor set. Thus, each node knows about the services of $n_d + 1$ nodes. Hence,

$$P_{\text{nomatch}} = \left(1 - \frac{P_s}{G_s \cdot I_s}\right)^{n_d+1}. \tag{18}$$

Combining Eqs. (17) and (18), the expression of $P_{k,\text{forward}}$, as shown in the theorem, is obtained. □

Theorem 4. *Compared with GSD, only by using BSU scheme in CNPGSDP, at each node that should forward a request packet, the average number of service request packets saved, N_s, is given by*

$$N_s = \sum_{k=2}^{n_d} ((k-1) \cdot P_{k,\text{forward}}). \tag{19}$$

Proof. The number of candidate nodes k ranges from 0 to n_d, which should be considered in two cases.

- If $k = 0$, both in GSD and in CNPGSDP, only one service request packet will be sent in broadcast mode. In this case, the number of saved request packets is 0.
- If $1 \leqslant k \leqslant n_d$, then in GSD, k unicast request packets will be sent, whereas in CNPGSDP, only by using BSU scheme, only one request packet transmitted in broadcast mode will be sent. Thus, $k - 1$ request packets are saved. According to Theorem 2, the probability of this case is $P_{k,\text{forward}}$.

Hence, the average number of saved request packets is

$$\begin{aligned} N_s &= 0 \cdot P_{0,\text{forward}} + \sum_{k=1}^{n_d} ((k-1) \cdot P_{k,\text{forward}}) \\ &= \sum_{k=1}^{n_d} ((k-1) \cdot P_{k,\text{forward}}) \\ &= (1-1) \cdot P_{1,\text{forward}} + \sum_{k=2}^{n_d} ((k-1) \cdot P_{k,\text{forward}}) \\ &= \sum_{k=2}^{n_d} ((k-1) \cdot P_{k,\text{forward}}). \end{aligned}$$ □

4.2. Verification through simulations

Theoretical results in Section 4.1 are verified through simulations in this section. Among the four assumptions proposed to facilitate the theoretical analysis in previous section, Assumptions 3 and 4 are not valid in simulations in common network simulators. Besides, node movement can distort simulation results from theoretical results further. Hence, verifications are performed in matlab instead of general network simulators.

Five groups of simulations are performed in matlab. In these five groups, G_s is set to 2, 4, 6, 8, and 10, respectively. Each simulation group consists of 50 simulations. Simulations are performed and results are operated as follows:

- Distribute 100 nodes uniformly in $1\,\text{m} \times 1\,\text{m}$ region. Assign each node to be a server with probability P_s. Assign each server a service randomly selected from.
- Generate a request searching for a service randomly selected from $G_s \times I_s$ services. Find each node w that nodes in $N_d(w) \cup \{w\}$ do not provide any matched services (denote the set of all founded nodes as F). Record the average number of candidate nodes for nodes in F.
- Average the recorded numbers over 50 simulations in each simulation group.

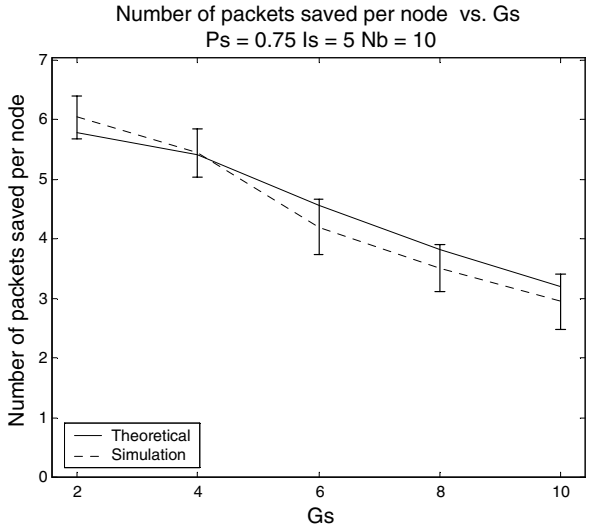


Fig. 6. Verification of Theorem 4.

Simulation results as well as theoretical values are shown in Fig. 6. Simulation results fit theoretical values well. The distortion may come from border effect.

5. Simulation results

To evaluate the improvement of CNPGSDP over GSD, these two protocols are implemented. We also implement flooding and FFPSDP for comparison. Recall that P represents the probability that a node providing no matched services will forward the packet. Flooding $P = 1$, referred as BASIC in the following text) is selected as a benchmark. In FFPSDP [17], probability P varies according to the following formula along with travel of request packets:

$$P = P_{F,\min} + (P_{F,\max} - P_{F,\min}) \cdot \frac{H_{\text{remain}}}{H_{\max}},$$

where $P_{F,\max}$ is the maximum value of the probability that a node will forward an unmatched request packet, $P_{F,\min}$ is the minimum value of the probability that a node will forward an unmatched request packet, H_{remain} is the value of the *remain-hop* field of the request packet to be forwarded, and $H_{\max}$ is the maximum number of hops that request packets can travel.

5.1. Performance metrics

Four performance metrics are considered in our simulations.

- *Request-Packet-Number*: It measures the number of service request packets sent in one simulation. It can exhibit the direct effect of BSU and CNP.
- *Succeeded-SDP-Number*: It is the number of SDP sessions in which the source has received at least one reply packet. It reflects the effectiveness (service discoverability) of service discovery protocols.
- *First-Response-Time*: It is the interval between the arrival of the first reply packet and the generation of the corresponding request packet. This metric is averaged over all succeeded SDP sessions. It measures the promptness of service discovery protocols. It also indirectly reflects the averaged distance between a client and the corresponding first replier.
- *Ratio of Succeeded-SDP-Number to Total-SDP-packet-number (Suc2Total)*: This metric is the ratio of Succeeded-SDP-number to the number of all request packets and reply packets. It reflects the efficiency of service discovery protocols. Although service advertisement packets in GSD and CNPGSDP make up a part of packet overhead, they are not included in total-SDP-packet since the number of these packets is greatly affected by protocol parameters. Additionally, omitting service advertisement packets helps to make a more discriminative comparison between GSD and CNPGSDP.

5.2. Two effects of service advertisement packet spreading

Service advertisement packet spreading operation has two effects server-manifold-effect and hop-shrink-effect.

5.2.1. Server-manifold-effect

Because of the service advertisement packet spreading operation, service information provided by a node can be cached by all nodes in its d-hop neighbor set. When receiving a request packet that needs the service, all of them can respond based on the cached information as if they all are servers. This effect is called as server-manifold-effect. This effect leads to more reply packets and larger succeeded-SDP-number.

5.2.2. Hop-shrink-effect

Because of the service advertisement packet spreading operation, nodes in a server's d-hop

neighbor set all know about the services provided by the servers. Thus, a service request will be matched in fewer hops. This effect is called as hop-shrink-effect. This effect can lead to fewer request packets and fewer reply packets. The reason is as follows. When matched, the request packet will not be forwarded any longer. Because of the hop-shrink-effect, the match is found at a node nearer to the source. Hence, compared with no service advertisement operation, the number of request packets will be reduced. Meanwhile, the reply packet can reach the corresponding source in few hops, and thus, reply packets are reduced also.

5.3. Simulation settings

Simulation studies are performed using Glomosim [25]. The distributed coordination function (DCF) 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 speed randomly selected $V \in [V_{\mathrm{MIN}}, V_{\mathrm{MAX}}]$. When reaching its destination, a node keeps static for a random period $T_P \in [T_{\mathrm{MIN}}, T_{\mathrm{MAX}}]$. When the period expires, the node will then randomly select a new destination and move to the new destination with new speed. The process will repeat permanently. In our simulations, $T_{\mathrm{MIN}} = T_{\mathrm{MAX}} = 0$, $V_{\mathrm{MIN}} = V_{\mathrm{MAX}}$.

Some basic parameters that are used in all the following simulations are set as shown in Table 3. Simulation scenarios are created with 100 nodes randomly distributed in the scenario area. At the beginning of each simulation, some nodes are randomly selected out to act as servers. These selected servers provide randomly selected services. During each simulation, 100 SDP sessions are started at randomly selected time by randomly selected nodes.

Table 3
Basic parameters

Parameters	Value
Scenario area	1000 m × 1000 m
Node number	100
Simulation time	1000 s
Wireless bandwidth	1 Mbps
SDP session number	100
$P_{F,\max}$ (FFPSDP)	1
$P_{F,\min}$ (FFPSDP)	0.4
Service advertisement interval	20 s
Valid time of SIC item	21 s
Number of service group	2
Number of service info in each group	5
Maximum hop of request packets	3
Maximum hop of advertisement packets	1

Node speed, radio range, and the number of servers are three major factors that affect the performance of service discovery protocols. In the following, we present the performance of the four service discovery protocols under the effects of these of factors. In all the following figures showing simulation results, error bars report 95% confidence.

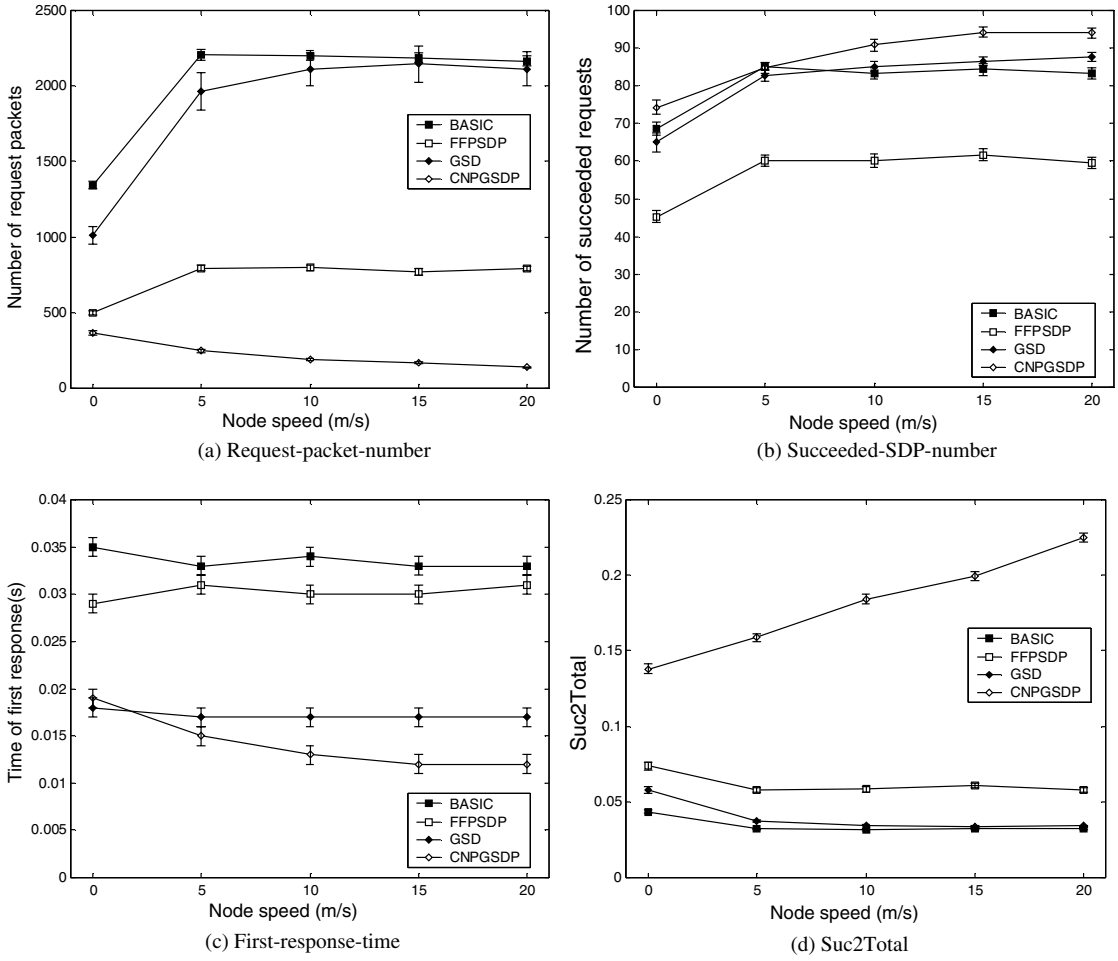
5.4. Effects of node speed

To inspect the effects of node speed, we run four simulation sets that use the four selected service discovery protocols, respectively. In these simulations, (1) radio range is set to 150 m, (2) the number of servers is fixed to 50. Each set includes five subsets of simulations, where $V = V_{\mathrm{MIN}} = V_{\mathrm{MAX}}$ and V is set to 0 m/s, 5 m/s, 10 m/s, 15 m/s, and 20 m/s, respectively. Each subset consists of 50 similar simulations. Simulation results are averaged over 50 simulations. The results are shown in Fig. 7.

Fig. 7(a) shows that, when node speed is 20 m/s, the request-packet-number metric of CNPGSDP is only about 6.5% of GSD, 17.4% of FFPSDP, and 6.3% of BASIC. CNPGSDP has the lowest service request packet overhead under different node speed. There are two reasons. First, BSU scheme reduces the number of service request packets. Second, CNP scheme reduces the number of candidate nodes, which lead to fewer request packets sent by the receivers. Thus, CNP reduces the number of service request packets further. This metric of CNPGSDP decreases gradually as node speed increases. This is because that node's movement expands the spreading range of service advertisement packets. Thus, hop-shrink-effect is enhanced. But for GSD, node movement leads to more candidate nodes at each node.

Fig. 7(b) shows the effect of node speed on the succeeded-SDP-number metric. The figure indicates that, although many service request packets are saved, CNPGSDP is still generally the most effective protocol under different node speed. Additionally, the superiority of CNPGSDP becomes more significant as node speed increases. This is because of more effective packet transmissions resulting from fewer request packets in CNPGSDP, which is more significant with higher speed.

Fig. 7(c) shows the effect of node speed on the first-response-time metric. CNPGSDP is the most prompt protocol under different node speed.



(a) Request-packet-number (b) Succeeded-SDP-number

(c) First-response-time (d) Suc2Total

Fig. 7. Effects of node speed on the performance of the service discovery protocols.

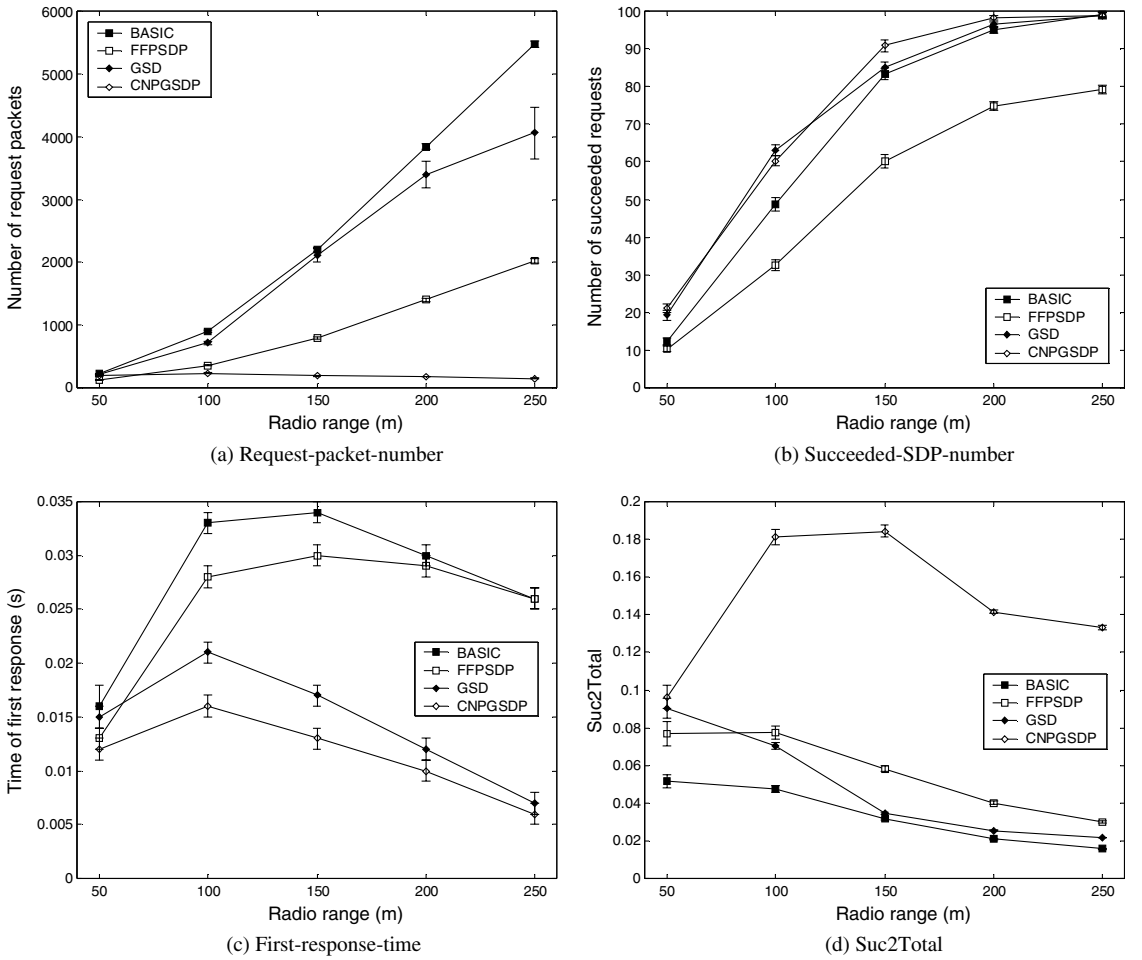
CNPGSDP and GSD are much superior to BASIC and FFPSDP in this metric due to the hop-shrink-effect. CNPGSDP outperforms GSD in this metric because of fewer packet collisions resulting from fewer packets in CNPGSDP.

Fig. 7(d) shows the effect of node speed on the Suc2Total metric. The superiority of CNPGSDP in this metric is more significant with higher node speed. When node speed is 20 m/s, the efficiency of CNPGSDP is about 6.5 times of GSD, 3.8 times of FFPSDP, and 7 times of BASIC.

5.5. Effects of radio range

Radio range is varied in simulate scenarios with different node densities. To inspect the effects of radio range, four other simulation sets are performed. In all these simulations, (1) node speed $V = V_{\mathrm{MIN}} = V_{\mathrm{MAX}}$ and V is fixed to 10 m/s, (2) the number of servers is set to 50. Each simulation set includes five subsets where radio range is set to 50 m, 100 m, 150 m, 200 m, and 250 m, respectively. Each subset consists of 50 similar simulations. The results are shown in Fig. 8.

Fig. 8(a) shows the effect of radio range on request-packet-number. The results show that CNPGSDP generally has the lowest service request packet overhead under different radio range. As radio range increases, the number of request packets in CNPGSDP keeps almost constant. The explanation is as follows. As radio range increases, the number of nodes covered by a SDP session will also increase. On the other hand, the hop-shrink-effect becomes more significantly. Therefore, as the joint effect, the number of request packets keeps almost constant. In contrast, as radio range increases, the



(a) Request-packet-number (b) Succeeded-SDP-number

(c) First-response-time (d) Suc2Total

Fig. 8. Effects of radio range on the performance of the service discovery protocols.

number of request packets in GSD increases quickly. This is due to more unicast request packets resulting from more candidate nodes.

Fig. 8(b) shows the effect of radio range on the succeeded-SDP-number metric. The results indicate that CNPGSDP is generally the most effective protocol under different radio range. The service discoverability of all these protocols increases as radio range increases. This is because of more larger coverage of request packets.

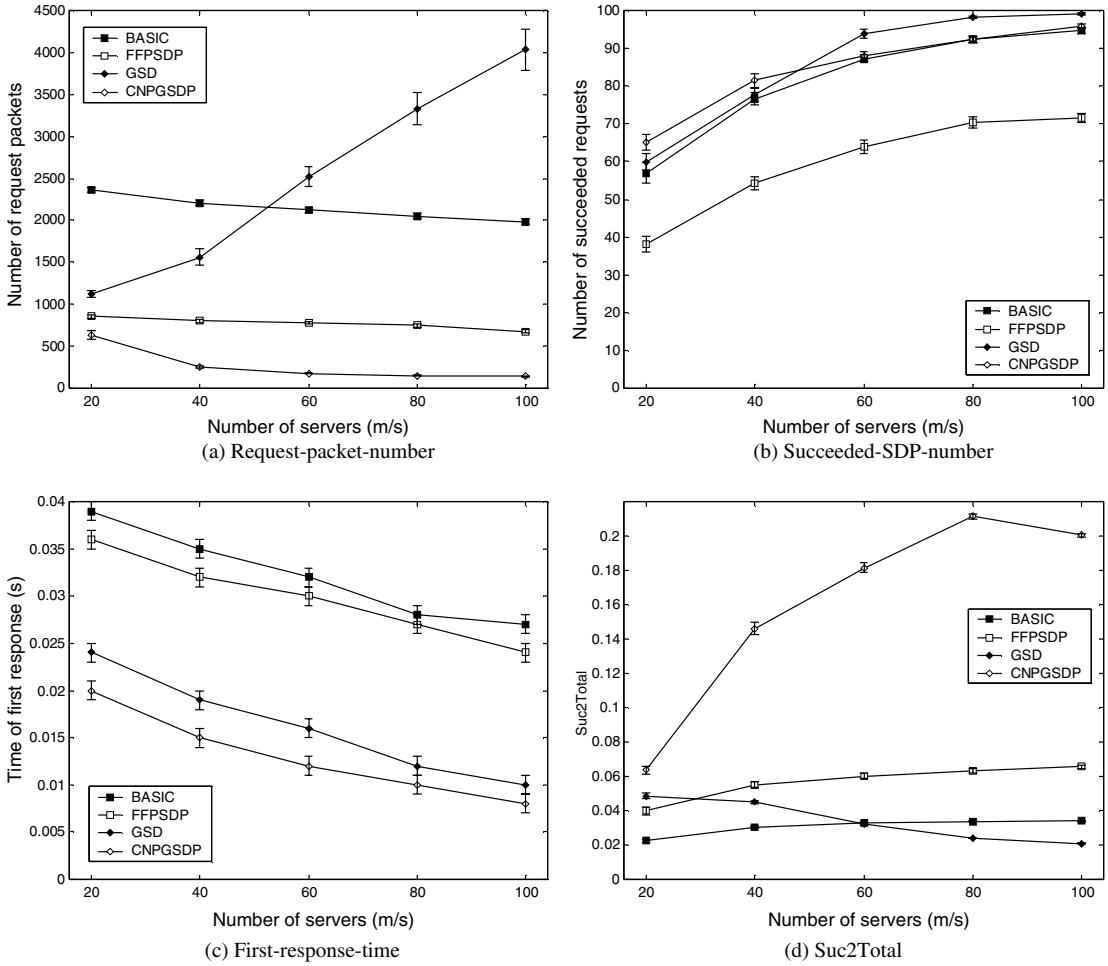
Fig. 8(c) shows the effect of radio range on the first-response-time metric. CNPGSDP is still the most prompt protocol under different radio range. The reason is the hop-shrink-effect and less collisions resulting from less packet overhead in CNPGSDP.

Fig. 8(d) shows the effect of radio range on the Suc2Total metric. The superiority of CNPGSDP in this metric is more significant with longer radio range. When radio range is 250m, the efficiency of CNPGSDP is about 6.1 times of GSD, 4.4 times of FFPSDP, and 6.5 times of BASIC.

5.6. Effects of number of servers

To inspect the effects of the number of servers, four other simulation sets are performed. In all these simulations, (1) node speed $V = V_{\mathrm{MIN}} = V_{\mathrm{MAX}}$ and V is fixed to 10 m/s, (2) radio range is set to 150 m. Each simulation set includes five subsets where the number of servers is set to 20, 40, 60, 80, and 100, respectively. Each subset consists of 50 similar simulations. The results are shown in Fig. 9.

Fig. 9(a) shows the effect of the number of servers on request-packet-number. CNPGSDP has the lowest request packet overhead under different number of servers. As the number of servers increases, the number of service request packets of all protocols



(a) Request-packet-number (b) Succeeded-SDP-number

(c) First-response-time (d) Suc2Total

Fig. 9. Effects of number of servers on the performance of the service discovery protocols.

except for GSD is reduced. This is because that service requests tend to be matched in fewer hops. Because of the hop-shrink-effect and server-manifold-effect, CNPGSDP is more sensitive to the change of server number than FFPSDP and BASIC. However, because of more unicast request packets, the number of service request packets in GSD increase quickly as the number of servers increases.

Fig. 9(b) shows the effect of number of servers on succeeded-SDP-number. CNPGSDP generally has the most powerful service discoverability. CNPGSDP is superior to GSD with fewer servers, whereas inferior to GSD with more servers because that protocol's service discoverability is less sensitive to protocol efficiency when there are plenty of matched servers in the network.

Fig. 9(c) shows the effect of the number of servers on the first-response-time metric. For the same reasons shown in previous explanations, CNPGSDP is the most prompt protocol under different number of servers. Response time of these protocols decreases as the number of servers increases. this is because that service requests tend to be matched in fewer hops.

Fig. 9(d) shows the effect of the number of servers on the Suc2Total metric. The superiority of CNPGSDP in this metric is more significant with more servers. When the number of servers is 100, the efficiency of CNPGSDP is about 9.6 times of GSD, 3 times of FFPSDP, and 6.0 times of BASIC.

In summary, from all these simulation results, we can see that CNPGSDP is generally the most effective, the most efficient, and the most prompt one among tested service discover protocols. Additionally, the superiority of CNPGSDP over other tested protocols is generally more significant with higher node speed, longer radio range, and larger number of servers.

6. Conclusions

In this paper, we proposed an efficient service discovery protocol for MANETs: Candidate Node Pruning enhanced Group-based Service Discovery Protocol (CNPGSDP). CNPGSDP introduces two schemes to enhance GSD: Broadcast Simulated Unicast (BSU) and Candidate Node Pruning (CNP). With BSU, several unicast packets in GSD is substituted by one broadcast packet that encloses all receivers. With CNP, the number of candidate nodes is reduced. Consequently, the number of successive request packets, which are the request packets sent by the receivers, is also reduced. By this means, CNP decreases the number of service request packets significantly.

Through mathematical analysis and simulations, we show that (1) CNPGSDP generally has the lowest packet overhead, (2) the efficiency of CNPGSDP can be several times of some typical service discovery protocols, and (3) the response time of CNPGSDP is much shorter than other tested protocols. In conclusion, CNPGSDP is a very effective, efficient, and prompt service discovery protocol for MANETs.

References

[1] IETF, Mobile ad-hoc network (MANET) working group, Mobile ad-hoc networks (MANET). [Online] Available from: <http://www.ietf.org/html.charters/manet-charter. html>.

[2] U.C. Kozat, L. Tassiulas, Service discovery in mobile ad hoc networks: an overall perspective on architecture choices and network layer support issues, Ad Hoc Networks 2 (2004) 23–44.

[3] E. Guttman, C. Perkins, J. Veizades, M. Day, Service location protocol, version 2, IETF RFC 2608, June 1999. [Online] Available from: <http://www.faqs.org/rfcs/rfc2608.html>.

[4] Sun Microsystems, Jini architecture specification, November 1999. [Online] Available from: <http://www.javasoft.com/products/jini/specs/jini-spec.pdf>.

[5] Microsoft Corporation, Universal plug and play: background. [Online] Available from: <http://www.upnp.org/resources/UpnPbkgnd.htm>.

[6] Salutation Consortium, Salutation architecture specification, 1999. [Online] Available from: <http://www.salutation.rog/specordr.htm>.

[7] CORBA, The common object request broker: architecture and specification, Version 2.6. [Online] Available from: <http://www.opengroup.org/infosrv/Brand/SPS_pdf/X01OB.pdf>.

[8] Bluetooth SIG, Bluetooth specification, Version 1.0B. [Online] Available from: <https://www.bluetooth.org/foundry/specification/document/Bluetooth_Core_10_B.pdf>.

[9] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, A. Schade, DEAPspace—transient ad hoc networking of pervasive devices, Computer Networks 35 (4) (2001) 411–428.

[10] M. Nidd, Service discovery in DEAPspace, IEEE Personal Communications 8 (4) (2001) 39–45.

[11] S. Motegi, K. Yoshihara, H. Horiuchi, Service discovery for wireless ad hoc networks, in: Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC'02), vol. 1, 2002, pp. 232–236.

[12] M. Barbeau, Service discovery protocols for ad hoc networking, in: Proceedings of the Workshop on Ad Hoc Communications (WAHC'00), 2000.

[13] P.E. Engelstad, Y. Zheng, R. Koodli, C.E. Perkins, Service discovery architectures for on-demand ad hoc networks, International Journal of Ad Hoc and Sensor Networks 1 (3) (2005).

[14] Y. Yuan, W. Arbaugh, A secure service discovery protocol for MANETs, in: Proceedings of the 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication Proceedings (PIMRC'03), Beijing, China, September 2003, vol. 1, pp. 502–506.

[15] L. Cheng, I. Marsic, Service discovery and invocation for mobile ad hoc networked appliances, in: Proceedings of the 2nd International Workshop on Networked Appliances (IWNA'00), New Brunswick, NJ, USA, December 2000.

[16] S. Helal, N. Desai, V. Verma, C. Lee, Konark—a service discovery and delivery protocol for ad-hoc networks, in: Proceedings of the 3rd IEEE Conference on Wireless Communication Networks (WCNC'03), New Orleans, USA, March 2003, pp. 2107–2133.

[17] Z.G. Gao, X.Z. Yang, S.B. Cai, Flexible forward probability based service discovery protocol for MANETs, Journal of Harbin Institute of Technology (Chinese) 37 (9) (2005) 1256–1260.

[18] Z.G. Gao, X.Z. Yang, T.Y. Ma, S.B. Cai, RICFFP: an efficient service discovery protocol for MANETs, in: Proceedings of the 2004 International Conference on Embedded And Ubiquitous Computing (EUC'04), Aizu-Wakamatsu City, Japan, Springer-Verlag, 2004, pp. 786–795.

[19] D. Chakraborty, A. Joshi, Y. Yesha, T. Finin, Towards distributed service discovery in pervasive computing environments, IEEE Transactions on Mobile Computing, in press.

[20] D. Chakraborty, A. Joshi, Y. Yesha, T. Finin, GSD: a novel group-based service discovery protocol for MANETs, in: Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN'02), Stockholm, Sweden, 2002, pp. 140–144.

[21] M. Klein, B.K. Ries, P. Obreiter, 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), 2003, pp. 180–185.

[22] M. Klein, M. Hoffman, D. Matheis, M. Mussig, Comparison of overlay mechanisms for service trading in ad hoc networks, Technical Report No. 2004-2, University of Karlsruhe, October 2004.

[23] Y.C. Tseng, S.Y. Ni, Y.S. Chen, J.P. Sheu, The broadcast storm problem in a mobile ad hoc network, ACM Wireless Networks 8 (2) (2002) 153–167.

[24] Y. Sasson, D. Cavin, A. Schiper, Probabilistic broadcast for flooding in wireless mobile ad hoc networks, Technical Report IC/2002/54, Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland, 2002, pp. 1–7.

[25] Wireless Adaptive Mobility Lab, Department of Computer Science, UCLA, Glomosim: a scalable simulation environment for wireless and wired network system. [Online] Available from: <http://pcl.cs.ucla.edu/projects/domains/glomosim.html>.

Zhenguo Gao received his BS and MS degree in Mechanical and Electrical Engineering from Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively. He is currently a Ph.D. candidate in Computer Science and Technology at Harbin Institute of Technology, Harbin, China. His research interests include service discovery, broadcasting, and routing for mobile ad hoc networks.

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 August 2003 to August 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, network survivability and security, switch scheduling and control, computer architectures, and embedded systems.

Xiaozong Yang is a Professor in School of Computer Science and Technology school of Harbin Institute of Technology. His current research interests are: computing architecture, fault tolerant computing, fault injection, wireless network.