Computing and Informatics, Vol. 29, 2010, 1283–1305

SIGIFSDP: A SERVICE ID GUIDED INTELLIGENT FORWARDING SERVICE DISCOVERY PROTOCOL IN PERVASIVE COMPUTING ENVIRONMENTS

Qianli Ma

School of Computer and Information Technology
Shanxi University
Taiyuan, 030006, China
e-mail: mymql123@163.com

Minghong Liao, Shouxu Jiang, Wenxu Xu

School of Computer Science and Technology
Harbin Institute of Technology
Harbin, 150001, China
e-mail: liao@hit.edu.cn, jsx@hit.edu.cn, xuwenxu@hit.edu.cn

Zhenguo Gao

College of Automation
Harbin Engineering University
Harbin, 150001, China
e-mail: gag@ftcl.hit.edu.cn

Manuscript received 18 November 2008; revised 13 July 2009
Communicated by Jacek Kitowski

Abstract. Service discovery constructs a bridge between the service providers and the service consumers, and is a key point in pervasive computing environments. In group-based service discovery protocols, selective forwarding service requests only based on the service group maybe lead to unnecessary forwarding, which produces large packet redundancy. This paper proposes an efficient service discovery protocol: SIGIFSDP (Service Id Guided Intelligent Forwarding Service Discovery Protocol).

In SIGIFSDP, based on GSD, SIGIF (Service Id Guided Intelligent Forwarding) is introduced to select the exact forwarding nodes based on the service id. Theoretical analysis and simulation results using GloMosim verify that SIGIFSDP can save the response time, reduce the service request packets, and improve the efficiency of service discovery.

Keywords: Pervasive computing, service discovery, group-based, IEEE 802.11

Mathematics Subject Classification 2000: 68M10

1 INTRODUCTION

In today's information society, the whole world is going wireless and mobile [1]; so pervasive computing environment is becoming tremendous popular. Pervasive computing environments include handheld, wearable, and embedded computers in addition to regular desktop clients and servers. These are interconnected using some combinations of wireless ad-hoc networks and wireless infrastructure-based networks, such as WLANs [2]. In such dynamic environments, the cohort of computing elements participating in any distributed system dynamically changes with time. Lack of fixed infrastructure support is a natural phenomenon in such environments, which leads to dependency on other devices for resources [3]. Efficient and timely service discovery is a prerequisite for good utilization of shared resources in the network. Therefore service discovery is an important aspect in pervasive computing environments. Service discovery is the technology of finding services matching one's needs in the networks [4], and thus it constructs a bridge between the service providers and the service consumers.

There have been considerable academic and industrial research efforts in service discovery in pervasive computing environments. Protocols like Jini [5], UPnP [6], UDDI [7] and Service Location Protocol [8] have been developed to discover services in the wired networks. The architectures followed by these protocols are primarily centralized/semi-centralized, registration-oriented and have an implicit assumption that the underlying network is stable and is capable of providing reliable communication. They are not suitable for wireless networks which are not stable. The service discovery protocols based on wireless networks also have a lot of problems. The flooding used in Konark [9] may lead to great packet redundancy, serious contention, and waste of bandwidth and energy. Lanes [10] and Service Rings [11] need to establish and maintain the multilayer hierarchical network structure, which results in extra cost. In GSD [12], PCPGSD [13] and CNPGSDP [14], services presented on nodes are classified into several groups. When forwarding a request packet, instead of broadcasting the request packet to all neighbors, these protocols, based on their service group, selectively forward the service request to those nodes where there are more chances of discovering matched services. However, this often leads to inaccurate match, and hence produces great packet overhead.

In this paper, we propose a novel service discovery protocol: SIGIFSDP (Service Id Guided Intelligent Forwarding Service Discovery Protocol). It inherits the virtues of group-based intelligent forwarding of service requests and peer-to-peer caching of service advertisements in GSD, and proposes the SIGIF (Service Id Guided Intelligent Forwarding) scheme. In SIGIF, a service advertisement packet contains not only the ids of service groups but also the ids of services. The service groups indicate the groups to which the services provided by the nodes in d-hop neighbor set of the corresponding source node (the advertisement sender) belong. So a service request will not be sent toward the nodes only matching the service group id, but to the nodes that both match the service group id and the service id. If no matching nodes could be found during intelligent forwarding of a service request, FFP (Flexible Forward Probability) [15] is used to transmit the service request.

The rest of this paper is organized as follows. In Section 2, a brief overview of GSD and PCPGSD is provided. In Section 3, SIGIFSDP is described in detail. Section 4 presents mathematical analysis on the forwarding probability of different protocols, which shows that SIGIFSDP can reduce packet overhead greatly. In Section 5, comparative studies between SIGIFSDP and three other service discovery protocols are performed through extensive simulations using GloMosim. Simulation results are shown, and in the end, a conclusion is given in Section 6.

2 REVIEWS OF GSD AND PCPGSD

GSD and PCPGSD are group-based distributed service discovery protocols in pervasive computing environments. They combine request-broadcast and service-advertisement together. Service advertisements are based on the concept of peer-to-peer caching. In light of the ids of service groups that the services provided by the nodes in the sender's vicinity belong to, service requests are selectively forwarded to those nodes in which there are more chances to discover the matched services.

In GSD, services are classified into several service groups based on their functionality, and each service group is composed of several different services. A server node advertises its services to the d-hop neighbor nodes in its vicinity. In addition to service descriptions of the sender, a service advertisement packet includes the list of service groups that the services provided by nodes in the d-hop neighbor set of the sender belong to. Whenever a node receives a service advertisement packet, it would store the packet in its SIC (Service Information Cache); and what it stores is the id of service group, not detailed service descriptions. The service group id is used to selectively forward service requests to those nodes that in their vicinity have seen the same group as the requested group. In this way, GSD can intelligently regulate the direction of forwarding service requests.

PCPGSD enhances GSD by three mechanisms: PFCN (Pruning of Far Candidate Nodes), CRN (Combining of Relay Nodes) and PRN (Piggybacking of Relay

Nodes). PFCN prunes all candidate nodes that are too far to be reached by the service requests. Instead of sending one service request to each candidate node, CRN selects a relay node to which the service request is transmitted. In place of unicasting several service requests, PRN sends only one service request in broadcast mode by piggybacking the list of relay nodes in the service request. Hence, PCPGSD can reduce packet overhead, and improve performance.

Despite the great value in GSD and PCPGSD, there are still a few aspects need to be considered carefully:

1. When a service request should be forwarded, based on the service groups in SIC, the current node will intelligently select the nodes whose neighbor nodes have the same service group as the requested group as the forwarding nodes. However, each service group includes several different services. When the service group meets the request's need, the concrete service often does not meet the request's demand. In such cases, the forwarding toward those nodes can not find matched service at last, which results in false forwarding. At this time, the service request may be forwarded to the nodes that their services belong to the requested service group, but do not match the requested service. It is obvious that false forwarding is unnecessary and should be avoided.
2. When a node receiving a service request can not find the forwarding nodes in terms of its SIC, it will broadcast the service request to its neighboring nodes with 100 % probability. However, 100 % broadcasting maybe leads to large duplicate service requests, especially when the maximum hop of service advertisements is smaller, which will greatly affect the protocol performance.

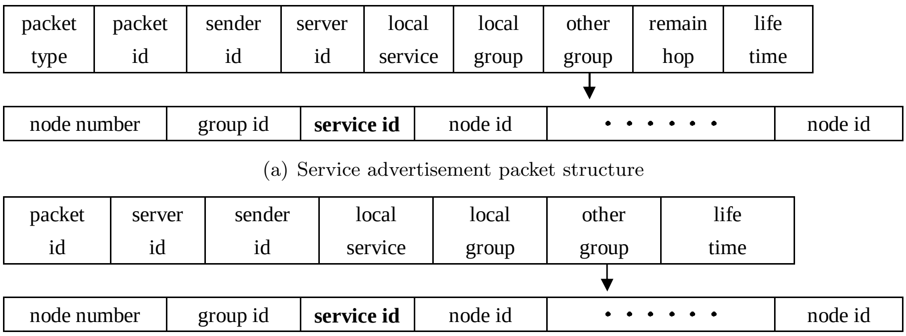
3 SERVICE ID GUIDED INTELLIGENT FORWARDING SERVICE DISCOVERY PROTOCOL (SIGIFSDP)

3.1 Data Structures

In comparison with PCPGSD, SIGIFSDP adds a *server id* subfield in the *other group* field of service advertisement packet and SIC to store the service id, which can help a node select the service-matched nodes as forwarding nodes. This modification can improve the protocol efficiency largely. The modified structures are shown in Figure 1. New fields are presented in boldface.

3.1.1 Service Advertisement Packet Structure

In SIGIFSDP, the structure of service advertisement packet is shown in Figure 1(a). In GSD and PCPGSD, a service advertisement packet contains the *other group* field, which encloses the list of service groups that the services provided by nodes in the *d*-hop neighbor set of the advertisement sender belong to. In SIGIFSDP, a service advertisement packet not only includes the *other group* field, but also adds the *server id* subfield to the *other group* field, in order to store the concrete service id contained in the service group. In this way, the service id, that the services provided by nodes in the d-hop neighbor set of the advertisement sender own, can be propagated to the receiving nodes with the service advertisement packet. It is remarkable that the *server id* field does not store detailed service descriptions, but just the service id. This method of storing only the service id just as the service group id would not increase the size of service advertisement packet greatly.



(b) SIC structure

Fig. 1. Modified Structure in SIGIFSDP

The *packet type* field is the type of message packet. The *packet id* field is used to distinguish different advertisement packets sent by one node, and increases monotonically with each service advertisement packet generated by the server node. The *sender id* field indicates the current sender of the packet. The *server id* field denotes the producer of service advertisement packet. The *local service* field stores the descriptions of services provided by the server node. The list of service groups that the local services belong to is stored in the *local group* field. The *other group* field contains the id of services provided by the server's d-hop adjacent nodes, the id of service groups that the services belong to, the list of nodes and the number of nodes. The value of the *remain hop* field indicates the number of hops that the service advertisement packet can still travel. The life time of service advertisement packet is stored in the *life time* field.

3.1.2 SIC Structure

In order to cache the corresponding contents of advertisement packets, SIGIFSDP also adds the *service id* subfield into the *other group* field of SIC. In light of the *service id* field, forwarding nodes can be selected accurately. While generating a new service advertisement packet, SIC must also be taken into account. The SIC structure is shown in Figure 1(b), each field of SIC is the same as that of service advertisement packet.

3.2 The SIGIF Scheme in SIGIFSDP

3.2.1 SIGIF Idea

While forwarding a service request, group-based service discovery protocols often select the forwarding nodes according to the SIC of current node, that is, select the SIC entry nodes which contain the same group as the requested group as the next hop nodes. Obviously, if the concrete services provided by those nodes can not match the requested service, transmitting the service request to those nodes is unnecessary and often leads to invalid matching. Because of forwarding only based on service group, it is inevitable that the service request packets should be discarded when the matched services can not be discovered meanwhile the hop number of packets reaches the maximum value. This results in a great waste of limited resources. Hence in SIGIFSDP, just like in GSD and PCPGSD, services are divided into several groups, and each service group contains several different services. However, the *other group* field of service advertisement packet and SIC is modified, the added *service id* subfield is used to store the id of services that the d-hop neighbor nodes own. When the forwarding nodes are selected, not only the service group id but also the service id must be considered carefully. Only in this way can the forwarding nodes be selected correctly. At this time, transmitting the service request to the selected nodes is right, which would achieve valid matching finally. So SIGIFSDP can reduce unnecessary packet forwarding, save network resources, and improve efficiency significantly.

It is certain that, comparing with selecting forwarding nodes only based on the service group id, selecting the forwarding nodes according to both the service group id and the service id often leads to fewer or no matching nodes. Consequently, there will be more service requests being sent with 100 % probability, which often produces large reduplicated service request packets, and therefore results in serious problems. So SIGIFSDP introduces the FFP scheme of FFPSDP [15] to calculate the forwarding probability P_f according to Equation (1). P_f is the current probability that one node will forward a service request packet, it would decrease linearly as the service request forwarded further, which not only retains the request's spreading range and spreading-ratio but also reduces the redundancy of service request packets. RH is the value in the *remain hop* field of the service request packet. MH is the value of the maximum hop number of the service request packet. Fixing FP_{Max} to 100 %, we may find a proper value for FP_{Min}. In this paper, FP_{Min} is defined as 40 %. With no forwarding nodes, transmitting the service request packet with P_f less than 100 % will achieve a more efficient usage of network bandwidth and power.

$$P_f = FP_{Min} + (FP_{Max} - FP_{Min}) \cdot \frac{RH}{MH} \tag{1}$$

Figure 2 shows a comparison of service request forwarding among SIGIFSDP, GSD and PCPGSD, respectively. In Figure 2, circles represent mobile nodes, the string in the circle indicates the identity of the node and the services it provides, e.g. string "*B, i_1*" means that the node is B and it provides a service "*i_1*", which belongs to the service group i. Double-headed arrows between two nodes indicate that the two nodes can communicate with each other directly. The thin-line arcs around a node denote unicasting a service request, and the thick-line arcs indicate broadcasting a service request. The thin-line table adjacent to arcs represents part contents of a service request packet, and the thick-line table adjacent to a node shows part contents of the node's SIC.

In Figure 2(b), using one broadcast packet instead of three unicast packets by storing the list of forwarding nodes in the service request packet, PCPGSD decreases the number of service request packets comparing with GSD in Figure 2(a). In Figure 2(c), the forwarding nodes can be selected exactly on the basis of the added *service id* field stored in the current node's SIC. Consequently, directly selecting the node D as the forwarding node avoids the service request's invalid transmitting to B and C, and saves the request packets. Figure 2 shows that the number of service request packets in SIGIFSDP, PCPGSD and GSD is 1, 3 and 5, respectively. Thus, SIGIFSDP has the least number of service request packets.

3.2.2 SIGIF Algorithm

In SIGIFSDP, the service request forwarding according to the service id must lead to finding matched services and producing a reply under an ideal condition, and so it is a deterministic matching. In GSD and PCPGSD, the service request forwarding based on the service group often results in invalid forwarding, and can not achieve a real matching at last, so it is a probability matching. The matching probability depends on the number of services in one service group. In SIGIFSDP, the forwarding algorithm of service request packet is shown as follows:

Algorithm: The Forwarding of Service Request Packet in SIGIFSDP
Input: *Req*: service request packet to be forwarded;
Variable Definitions:
  self_id the identity of current node;
  entry the SIC entry of current node;
  MAX_GROUP_NUM the maximum value of the other group in SIC of current node;
Output: updated and forwarded service request packet *Req* or produced service reply packet *Reply*;
Begin:
  Req. remainhop−−;
  if (*Req. remainhop* > 0) then
  {
    if (Matchedservice(*self_id*) == *Req. requestedservice*) then
    { // The current node is just the matching node,
      // creates and unicasts a service reply packet.
      CreatedandUnicast(*Reply*);

```
    }
    else if (Founditemnum(entry) == 0) then
    { // The current node has not any SIC item,
      // then modifies and broadcasts the service request.
      UpdatedandBroadcast(Req);
    }
    else
    { // Search for forwarding nodes in all of the SIC items of
      // current node, insert the results into the forwarding node
      // list, modify and forward the service request.
      while (Founditem(entry) <> NULL)
      {
        for(i=0; i<=MAX_GROUP_NUM; i++)
        {
          if (entry.lifetime<> 0 &&
            entry.othergroup[i].groupid==Req. requestedgroup &&
            entry.othergroup[i].serviceid==Req. requestedservice) then
          {
            for(j=0; j<=entry.othergroup[i].nodenumber-1; j++)
            {
              InsertnodeintoReceiverlist(entry.othergroup[i].nodeid[j]);
            }
          }
        }
        entry++;
      }
      UpdatedandForward(Req);
    }
  }
  else
  {
    DiscardPacket(Req);
  }
End.
```

4 MATHEMATICAL ANALYSIS

4.1 Theoretical Analysis

Some symbols used in mathematical analysis are listed in Table 1.

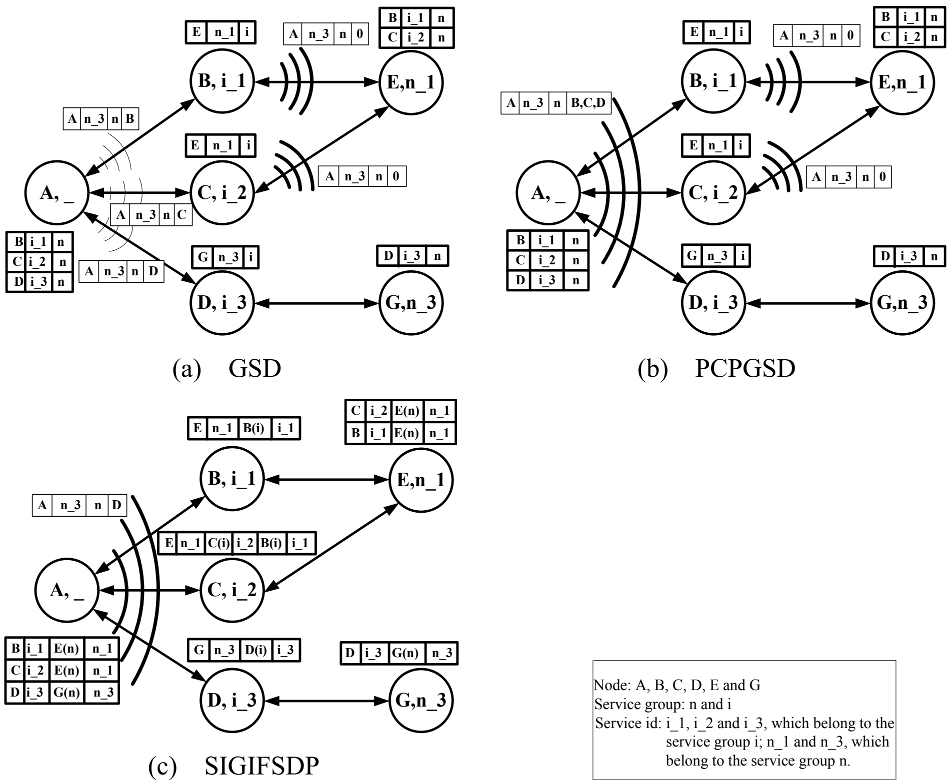


Fig. 2. Comparison of service request forwarding

Theorem 1. In SIGIFSDP, the probability that the current node will go on forwarding the request packet after receiving a service request packet is given by

$$P_{NEW} = \left(1 - \frac{P_S}{G_S I_S}\right) \cdot \sum_{i=1}^{n_d} C_{n_d}^{i} \cdot \left(\frac{P_S}{G_S I_S}\right)^{i} \cdot \left(1 - \frac{P_S}{G_S I_S}\right)^{(n_d - i)}. \tag{2}$$

Proof. In SIGIFSDP, if a node receiving a service request would go on forwarding the service request, it must be in two conditions as follows: (1) The current node has unmatched services, and in its neighboring nodes there is at least one node whose SIC contains the same service as the required service. It is denoted by P_1. (2) The current node has not any service, but in its neighbor nodes there is at least one node whose SIC contains the same service as the required service. It is expressed by P_2. Obviously,

$$P_{NEW} = P_1 + P_2. \tag{3}$$

Symbol	Denotation
P_S	The probability that a node has any service
G_S	The number of service groups
I_S	The number of services in each service group
d	The maximum hop number that the service advertisement packets can travel
n_d	The average number of nodes that are at most d-hops away from a node (excluding the node itself)
$C_{n_d}^i$	The number of combinations of selecting any i nodes from n_d nodes

Table 1. Symbols used in mathematical analysis

We calculate P_1 first, it can be presented as

$$P_1 = P_{1,1} \cdot P_{1,2} \tag{4}$$

where $P_{1,1}$ is the probability that the current node provides unmatched services. Because there are G_S service groups and I_S services in each group, the total number of services is $G_S I_S$. Since the services provided by the current node do not match the requested service, the services must be among $(G_S I_S - 1)$ services. Thus, we have

$$P_{1,1} = P_S \cdot \left(\frac{G_S I_S - 1}{G_S I_S} \right). \tag{5}$$

$P_{1,2}$ is the probability that, in the current node's d-hop neighbor set, there is at least one node whose SIC includes the service matching the requested service. That is, there are i nodes (i from 1 to n_d) in the d-hop neighbor set n_d of the receiving node, the nodes have matched services, and the other $n_d - i$ nodes have no matched services or have not any service. So we get

$$P_{1,2} = \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{1}{G_S I_S} \right)^i \cdot \left(P_S \cdot \frac{G_S I_S - 1}{G_S I_S} + 1 - P_S \right)^{(n_d - i)}. \tag{6}$$

Substituting $P_{1,1}$ and $P_{1,2}$ in Equation (4) with Equations (5) and (6), respectively, we can get

$$P_1 = P_S \cdot \left(\frac{G_S I_S - 1}{G_S I_S} \right) \cdot \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{1}{G_S I_S} \right)^i \cdot \left(P_S \cdot \frac{G_S I_S - 1}{G_S I_S} + 1 - P_S \right)^{(n_d - i)}. \tag{7}$$

Now we calculate P_2 as follows:

$$P_2 = P_{2,1} \cdot P_{2,2} \tag{8}$$

where $P_{2,1}$ is the probability that a node has not any service. Obviously,

$$P_{2,1} = 1 - P_S. \tag{9}$$

$P_{2,2}$ is the probability that, in the neighbor nodes of a node, there is at least one node whose SIC contains the same service as the requested service. Hence, $P_{2,2}$ is equal to $P_{1,2}$. Then

$$P_{2,2} = \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{1}{G_S I_S}\right)^i \cdot \left(P_S \cdot \frac{G_S I_S - 1}{G_S I_S} + 1 - P_S\right)^{(n_d - i)}. \tag{10}$$

Now, substituting $P_{2,1}$ and $P_{2,2}$ in Equation (8) with Equations (9) and (10), respectively, we have

$$P_2 = (1 - P_S) \cdot \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{1}{G_S I_S}\right)^i \cdot \left(P_S \cdot \frac{G_S I_S - 1}{G_S I_S} + 1 - P_S\right)^{(n_d - i)}. \tag{11}$$

So, based on P_1 in Equation (7) and P_2 in Equation (11), the following equation can be deduced:

$$P_{NEW} = \left(1 - \frac{P_S}{G_S I_S}\right) \cdot \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(\frac{P_S}{G_S I_S}\right)^i \cdot \left(1 - \frac{P_S}{G_S I_S}\right)^{(n_d - i)}.$$

Now the mathematical expression of P_{NEW}, as shown in Theorem 1, is obtained. □

Theorem 2. In GSD and PCPGSD, the probability that the current node will go on forwarding the request packet after receiving a service request packet is given by:

$$P_{OLD} = P_S \cdot \frac{I_S - 1}{G_S I_S} + \left(1 - \frac{P_S}{G_S}\right) \cdot \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{I_S - 1}{G_S I_S}\right)^i \cdot \left(1 - \frac{P_S}{G_S}\right)^{(n_d - i)}. \tag{12}$$

Proof. In GSD and PCPGSD, if a receiving node would go on forwarding the service request, it must be in three conditions as follows: (1) The current node has at least one unmatched service that belongs to the same group as the requested group. It is denoted by P_x. (2) The current node has at least one unmatched service that does not belong to the same group as the requested group, but in its neighbor nodes there is at least one node whose SIC contains the same service group as the requested group. It is expressed by P_y. (3) The current node has not any service, but in its neighbor nodes there is at least one node whose SIC contains the same service group as the requested group. It is expressed by P_z. Obviously,

$$P_{OLD} = P_x + P_y + P_z. \tag{13}$$

In the first case, P_x can be calculated as follows:

$$P_x = P_{x,1} \cdot P_{x,2} \tag{14}$$

where $P_{x,1}$ is the probability that the current node has one unmatched service that belongs to the same group as the requested group, so the service must be among

$I_S - 1$; thus we have

$$P_{x,1} = P_S \cdot \frac{I_S - 1}{G_S I_S}. \tag{15}$$

Because all d-hop neighbor nodes of the current node can receive the advertisement from it, and its service group matches the required service group, it is necessary that SIC of each its neighbor node contains an item matching the required service group, hence this node will surely forward the service request. In a word, if a node contains the services belonging to the same group as the requested group, it will certainly forward the service request. So

$$P_{x,2} = 1. \tag{16}$$

Substituting $P_{x,1}$ and $P_{x,2}$ in Equation (14) with Equations (15) and (16), we can get

$$P_x = P_S \cdot \frac{I_S - 1}{G_S I_S}. \tag{17}$$

P_y can be calculated as follows:

$$P_y = P_{y,1} \cdot P_{y,2} \tag{18}$$

where $P_{y,1}$ is the probability that the current node contains unmatched service, and the service does not belong to the same group as the requested group; then its service must be among $G_S I_S - I_S$ and thus we have

$$P_{y,1} = P_S \cdot \frac{G_S I_S - I_S}{G_S I_S}. \tag{19}$$

$P_{y,2}$ is the probability that in its neighbor nodes there is at least one node whose SIC contains the same service group as the requested group. That is, there is at least one node (such as i nodes) in the d-hop neighbor set n_d of the receiving node, and this node has matched the service group; the other $n_d - i$ nodes have no matched service groups or no service. So we get

$$P_{y,2} = \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{I_S - 1}{G_S I_S}\right)^i \cdot \left(P_S \cdot \frac{G_S I_S - I_S}{G_S I_S} + 1 - P_S\right)^{(n_d - i)}. \tag{20}$$

Now, substituting $P_{y,1}$ and $P_{y,2}$ in Equation (18) with Equations (19) and (20), we have

$$P_y = P_S \cdot \left(\frac{G_S I_S - I_S}{G_S I_S}\right) \cdot \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{I_S - 1}{G_S I_S}\right)^i \cdot \left(P_S \cdot \frac{G_S I_S - I_S}{G_S I_S} + 1 - P_S\right)^{(n_d - i)}. \tag{21}$$

Then, P_z can be calculated as follows:

$$P_z = P_{z,1} \cdot P_{z,2} \tag{22}$$

where $P_{z,1}$ is the probability that the current node has not any service, obviously

$$P_{z,1} = 1 - P_S. \tag{23}$$

$P_{z,2}$ is the probability that in its neighbor nodes there is at least one node whose SIC contains the same service group as the requested group. So, we can get $P_{z,2} = P_{y,2}$, that is

$$P_{z,2} = \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{I_S - 1}{G_S I_S}\right)^i \cdot \left(P_S \cdot \frac{G_S I_S - I_S}{G_S I_S} + 1 - P_S\right)^{(n_d - i)}. \tag{24}$$

Substituting $P_{z,1}$ and $P_{z,2}$ in Equation (22) with Equations (23) and (24) we can get

$$P_z = (1 - P_S) \cdot \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{I_S - 1}{G_S I_S}\right)^i \cdot \left(P_S \cdot \frac{G_S I_S - I_S}{G_S I_S} + 1 - P_S\right)^{(n_d - i)}. \tag{25}$$

So, based on P_x in Equation (17), P_y in Equation (21), and P_z in Equation (25), the following equation can be derived:

$$P_{OLD} = P_S \cdot \frac{I_S - 1}{G_S I_S} + \left(1 - \frac{P_S}{G_S}\right) \cdot \sum_{i=1}^{n_d} C_{n_d}^i \cdot \left(P_S \cdot \frac{I_S - 1}{G_S I_S}\right)^i \cdot \left(1 - \frac{P_S}{G_S}\right)^{(n_d - i)}.$$

Now the mathematical expression of P_{OLD}, as shown in Theorem 2 is obtained. □

4.2 Parameter Study

Theoretical results in Theorems 1 and 2 are compared by a series of experiments. In these experiments, we set different values for the sets of theorem parameters, and compute their results. The results show that the service request forwarding probability in SIGIFSDP is much less than that in GSD and PCPGSD. Thus it is verified that SIGIFSDP can produce fewer service request forwarding packets; so it decreases the packet overhead largely. Due to the space limitation in the paper, we only show two of these experiments in Figure 3. In Figure 3, the number of service groups G_S is fixed to 3, P_S is set to 0.3, n_d is equal to 5 or 10, the number of services I_S in each service group is set to 3, 4, 5, 6, 7, 8 and 9, respectively.

5 SIMULATION STUDIES

5.1 Definition of Performance Criterions

In our simulations, four criterions are studied. They are:

Definition 1 (Number of request packets.)**.** It indicates the average number of service request packets transmitted during one session. It stands for a part of the packet overhead.

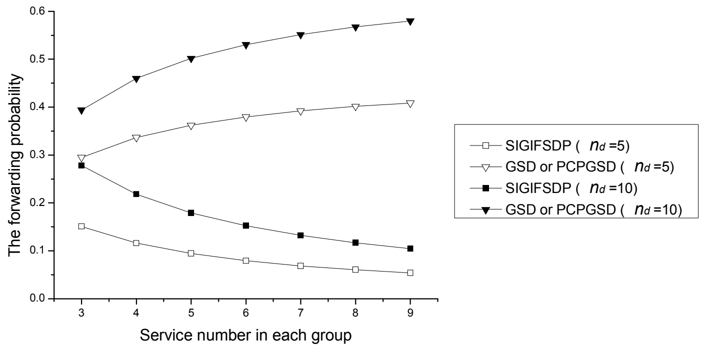


Fig. 3. Comparison of the service request forwarding probability

Definition 2 (Reply request ratio.)**.** It is the ratio of the number of service request packets to the number of reply packets in each session. It reflects direct effectiveness of a service discovery protocol.

Definition 3 (Discovery efficiency.)**.** It is the ratio of the average number of sessions in which at least one reply packet is received by the request node to the sum of the request packets and the reply packets in one session. It indicates the protocol efficiency.

Definition 4 (Average first response time.)**.** It is the interval of the average time between the arrival of the first reply packet and the generation of the corresponding request packet. It measures the promptness of service discovery protocols. It is indirectly affected by the distance between a requester and the corresponding first replier.

5.2 Simulation Settings

To perform comparative simulation analysis, four protocols are implemented in GloMosim [16]: the flooding service discovery protocol (denoted as BASIC), GSD, PCPGSD and SIGIFSDP. The Distributed Coordination Function (DCF) of IEEE 802.11 is used as the underlying MAC protocol. The Random Waypoint Model (RWM) is used as the mobility model. In this model, a node selects a random destination point and then moves towards the destination at a speed V selected randomly. After reaching the destination, it will keep static for a random period P. Then the node will randomly select a new destination and move to it at a new speed. The node will repeat the process continuously. In our experiments, $P = 0$.

5.3 Simulation Results

The simulation scenarios are created with 100 nodes randomly distributed in the scenario area, and among them, 50 nodes are selected randomly as servers at the beginning of each simulation to produce the services. In each simulation, 50 service discovery sessions will be set up at random time by randomly selected nodes. The basic parameters used in all following experiments are listed in Table 2. The parameter choice is primarily based on three aspects as follows: the realistic scenarios, the characteristics of SIGIFSDP protocol and the parameter choice of similar protocols for comparison.

Parameter	Value	Parameter	Value
Scenario area	1 000 m × 1 000 m	Service advertisement interval	20 s
Node number	100	Valid time of SIC item	21 s
Server node number	50	Wireless bandwidth	1 Mbps
Simulation time	1 000s	Session number in each simulation	50

Table 2. Basic parameters used in simulations

5.3.1 The Effects of the Maximum Hop of Request Packets

To study the effects of the maximum hop of request packets, we perform 4 simulation sets that simulate four service discovery protocols, respectively. In these simulations, the maximum hop of advertisement packets is set to 3, the node speed V is 0, the number of service groups is 2, and the number of services in each service group is 7. Each set includes 4 parts, in which the maximum hop of request packets is fixed to 1, 2, 3 and 4. Each simulation includes 50 sessions. The experiment results are shown in Figure 4.

Figure 4(a) shows the effects of maximum hop of request packets on the number of request packets. For each protocol, the number of request packets increases as the maximum hop of request packets increases from 1 to 4. However, in PCPGSD and SIGIFSDP, the slope is much less than that in BASIC and GSD. SIGIFSDP has the lowest number of request packets among the four service discovery protocols. With the maximum hop of request packets increasing, the advantage in SIGIFSDP is greater. This is because in SIGIFSDP, increasing the maximum hop of request packets gives more chances to find the exact forwarding nodes, and hence, more request packets can be saved.

Figure 4(b) shows the effects of maximum hop of request packets on the reply request ratio. Except for the point in which the maximum hop of request packets is 1, SIGIFSDP has the highest reply request ratio compared with all the other protocols. This indicates that the request packets are traveled too near to show SIGIFSDP characteristics in this point.

Figure 4(c) shows the effects of maximum hop of request packets on the discovery efficiency. The discovery efficiency decreases as the maximum hop of request packets increases. In PCPGSD and SIGIFSDP, the decrease is more obvious. Because PCPGSD creates more successful sessions which are actually invalid when the maximum hop is larger, based on the definition of discovery efficiency, the difference between PCPGSD and SIGIFSDP becomes small. SIGIFSDP has the highest discovery efficiency, especially when the maximum hop of request packets is 1, because the sum of request and reply packets is much less in this point in SIGIFSDP, the metric of SIGIFSDP is about 2.6 times of PCPGSD, 17.4 times of GSD, and 21.9 times of BASIC.

Figure 4(d) shows the effects of maximum hop of request packets on the average first response time. Even though SIGIFSDP has the highest discovery efficiency, it is still the most prompt protocol under different maximum hops of request packets. It is because the SIGIF scheme forwards service requests more exactly, and hence saves the response time.

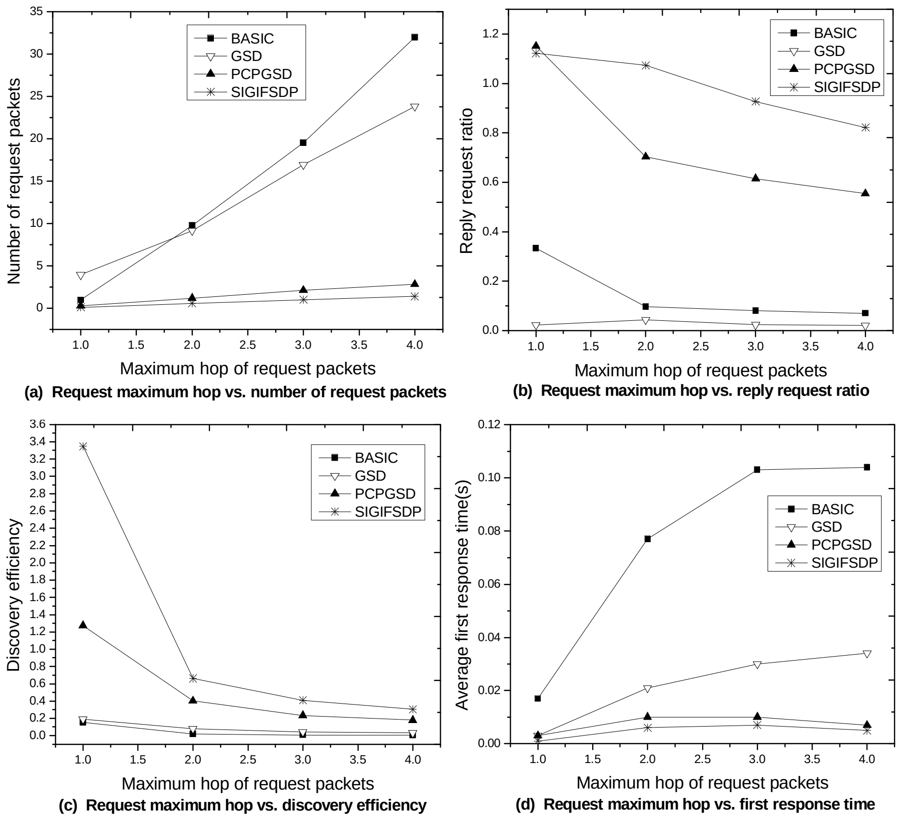


Fig. 4. Influence of the maximum hop of request packets

5.3.2 The Effects of the Maximum Hop of Advertisement Packets

Four simulation sets are run to study the effects of the maximum hop of advertisement packets on different protocols. In these simulations, the maximum hop of service requests is set to 3, the node speed V is 0, the number of service groups is 2, and the number of services in one service group is 7. Each set includes 5 parts, in which the maximum hop of advertisement packets is fixed to 1, 2, 3, 4 and 5. Each simulation is comprised of 50 sessions. The simulation results are exhibited in Figure 5.

Figure 5(a) shows the effects of maximum hop of advertisement packets on the number of request packets. PCPGSD and SIGIFSDP have fewer request packets than BASIC and GSD. SIGIFSDP has the least number of request packets among the four service discovery protocols, and its number of request packets reduces as the maximum hop of advertisement packets increases. This is because in SIGIFSDP, the bigger the maximum hop of advertisement packets is, the more nodes cache the server advertisement information in their SIC, there will be more chances to find the exact forwarding nodes, and hence, the more request packets can be saved.

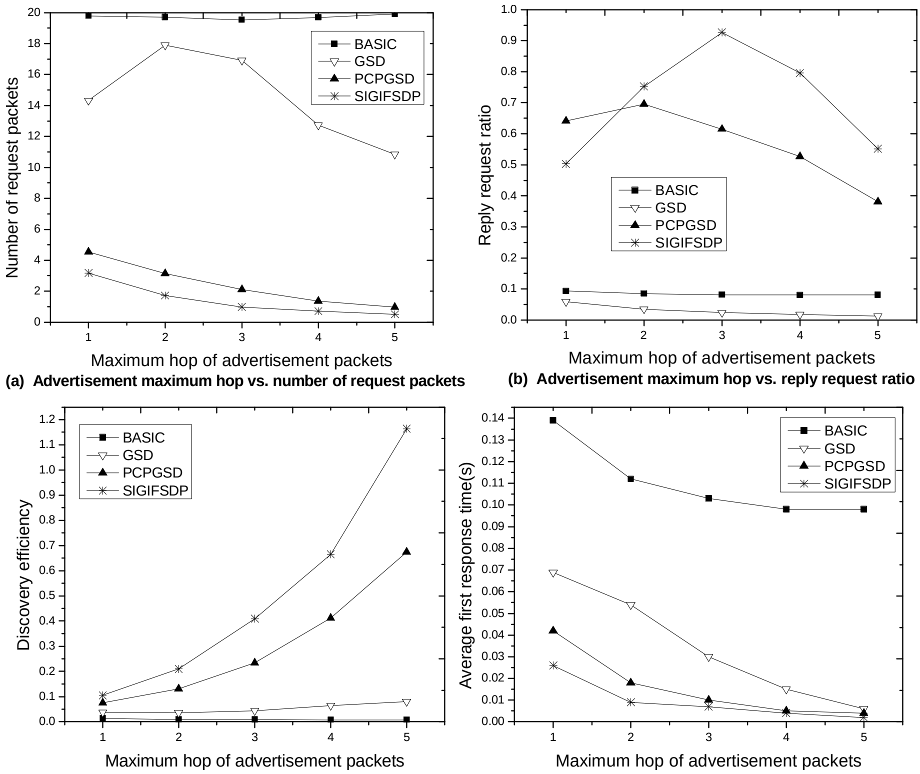
Figure 5(b) shows the effects of maximum hop of advertisement packets on the reply request ratio. Except for the point in which the maximum hop of advertisement packets is 1, SIGIFSDP has the highest reply request ratio in all the other points compared with other three protocols. This indicates that the advertisement packets are traveled so shortly that only few nodes store the advertisement information, so the reply request ratio is lower in this point.

Figure 5(c) shows the effects of maximum hop of advertisement packets on the discovery efficiency. Except for BASIC, the discovery efficiency increases as the maximum hop of advertisement packets increases. In PCPGSD and SIGIFSDP, the increase is more obvious. SIGIFSDP has the highest discovery efficiency. Especially when the maximum hop of advertisement packets is 5, the metric of SIGIFSDP is about 1.7 times of PCPGSD, 14.5 times of GSD, and 162.6 times of BASIC.

Figure 5(d) shows the effects of maximum hop of advertisement packets on the average first response time. In four different protocols, the average first response time decreases as the maximum hop of advertisement packets increases. In SIGIFSDP, when the maximum advertisement hop is bigger, because there are more nodes storing the advertisement information to SIC, the forwarding direction of request packets can be guided definitely based on the SIC, and hence the response time is saved. So SIGIFSDP has the shortest average first response time in all protocols.

5.3.3 The Effects of the Number of Services

Four simulation sets are run to study the effects of the number of services in each service group on different protocols. In these simulations, the maximum hop of service requests is set to 3, the node speed V is 0, the number of service groups is 3, and the maximum hop of advertisement packets is 3. Each set includes 4 parts, in which the number of services in each service group is fixed to 3, 5, 7 and 9, respectively. Each simulation includes 50 sessions. The simulation results are displayed in Figure 6.



(a) Advertisement maximum hop vs. number of request packets
(b) Advertisement maximum hop vs. reply request ratio
(c) Advertisement maximum hop vs. discovery efficiency
(d) Advertisement maximum hop vs. first response time

Fig. 5. Influence of the maximum hop of advertisement packets

Figure 6(a) shows the effects of the number of services on the number of request packets. For each protocol, the number of request packets increases with the number of services increasing; but for BASIC, the changing is trivial. This is because that the request forwarding is only based on group in PCPGSD and GSD, the more services are contained in each service group, it is more probable that the forwarding is invalid while looking for certain services; hence the more useless request packets will be forwarded. In SIGIFSDP, because of the request forwarding based on the service id, the bigger the number of services is in each service group, the fewer forwarding nodes will be discovered while looking for certain services; hence there will be more current nodes broadcasting the request packets with probability less than 100 %, which creates more request packets.

Figure 6(b) shows the effects of the number of services on the reply request ratio. PCPGSD and SIGIFSDP have higher reply request ratio, SIGIFSDP is superior to the other three protocols. For each protocol, the reply request ratio cuts down as the number of services increases. This is because the number of request packets increases with the number of services increasing. In Figure 6 a), based on the definition of reply request ratio, it is necessary that the reply request ratio decreases as the service number increases.

Figure 6(c) shows the effects of the number of services on the discovery efficiency. The discovery efficiency reduces as the number of services increases. However, in PCPGSD and SIGIFSDP, the slope is more than that in BASIC and GSD. SIGIFSDP has the highest discovery efficiency. Especially when the number of services in each group is 3, the metric of SIGIFSDP is about 1.5 times of PCPGSD, 8.1 times of GSD, and 57.6 times of BASIC.

Figure 6(d) shows the effects of the number of services on the average first response time. Among the four different protocols, SIGIFSDP has the most prompt response time, because in SIGIFSDP, the forwarding direction of request packets can be guided directly based on the service id stored in SIC, which makes the finding hops reduced, and then the response time is saved. Thus the average first response time is least in SIGIFSDP.

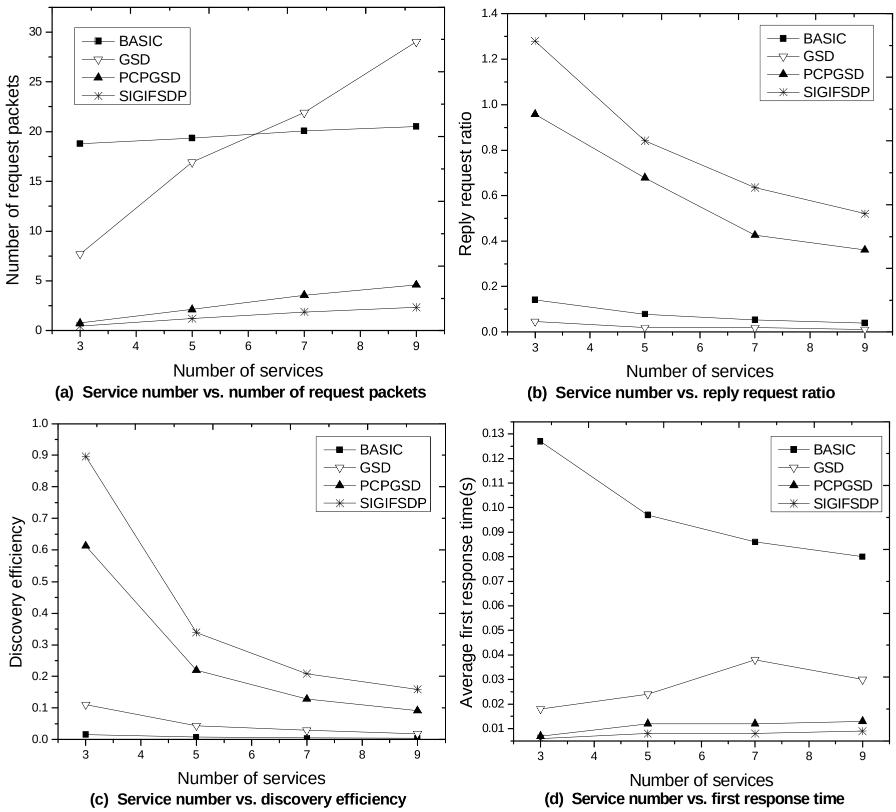


Fig. 6. Influence of the number of services in each service group

5.3.4 The Effects of the Node Speed

Four simulation sets are run respectively to study the effects of the node speed on different protocols. In these simulations, the maximum hop of service requests is set to 3, the maximum hop of advertisement packets is 3, the number of service groups is 2, and the number of services in each service group is 7. Each set includes 4 parts, in which the node speed is fixed to 3, 6, 9, 12 and 15, respectively. Each simulation includes 50 sessions. The simulation results are displayed in Figure 7.

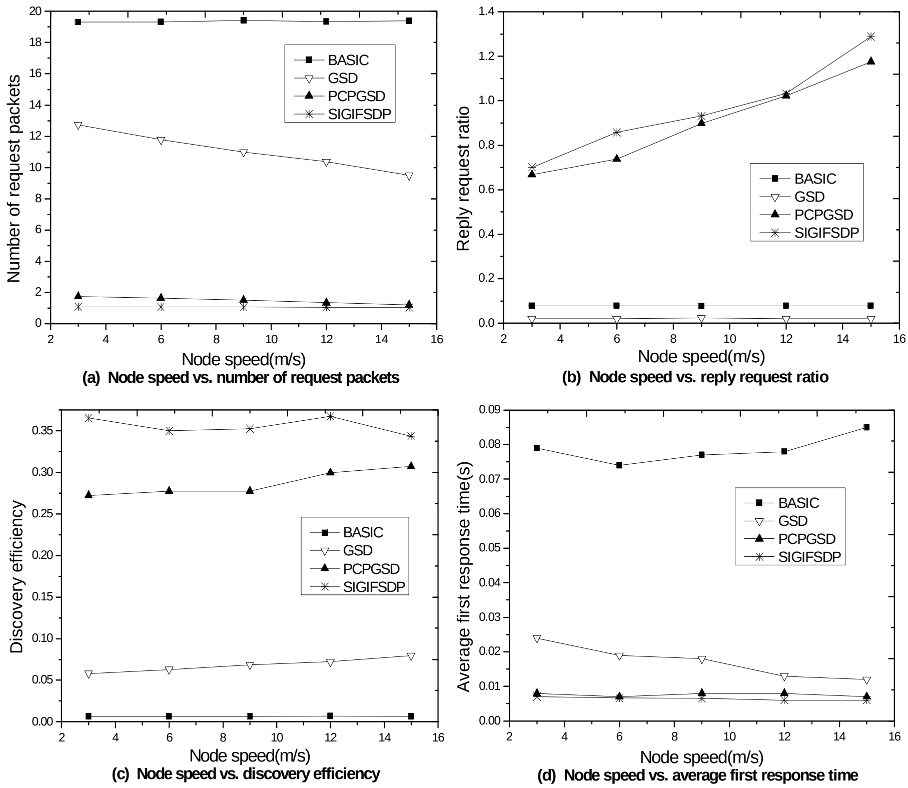


Fig. 7. Influence of the node speed

Figure 7(a) shows the effects of node speed on the number of request packets. SIGIFSDP has the lowest number of request packets among the four service discovery protocols. Except for BASIC, the number of request packets decreases as the node speed increases. This is because there are service advertisement operations in GSD, PCPGSD and SIGIFSDP. When a node moves to a new place, it will find new neighbors; so some new information will be cached in the node's SIC. Hence on the basis of more information, a request packet will be matched with higher probability at the node; so more request packets will be matched in fewer hops, and request packets can be saved.

Figure 7(b) shows the effects of node speed on the reply request ratio. PCPGSD and SIGIFSDP have higher reply request ratio, SIGIFSDP is superior to the other three protocols. This is because SIGIFSDP can forward the service request packets intelligently, and so it can find the matched nodes exactly. Meanwhile SIGIFSDP has the lowest number of request packets, based on the definition of reply request ratio, it is necessary that SIGIFSDP has the largest reply request ratio.

Figure 7(c) shows the effects of node speed on the discovery efficiency. Based on the definition of discovery efficiency, it is intelligible that SIGIFSDP has the highest discovery efficiency. Especially when the node speed is 3, the metric of SIGIFSDP is about 1.34 times of PCPGSD, 6.28 times of GSD, and 57.14 times of BASIC.

Figure 7(d) shows the effects of node speed on the average first response time. Even though SIGIFSDP has the highest discovery efficiency, it is still the most prompt service discovery protocol under different node speeds. It is because the SIGIF scheme forwards the service request more directly and exactly, and hence saves the response time.

6 CONCLUSIONS

In this paper, a novel service discovery protocol SIGIFSDP is introduced, which inherits the merits from GSD, proposes the scheme SIGIF to forward the service request toward the nodes that match both the service group id and the service id, not those that match only the service group id. When no forwarding nodes can be found, instead of broadcasting the service request with 100 % probability, SIGIFSDP forwards the service request with probability less than 100 %.

Mathematical analysis and simulation results both show that SIGIFSDP is superior to BASIC, GSD and PCPGSD in terms of service response time, packet overhead and service discovery efficiency. Therefore, it is certain that SIGIFSDP is an efficient, prompt service discovery protocol for pervasive computing environments.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 6070-3090). The authors thank Xin Gao and Guodong Sun for the early suggestions on this paper. We would also like to thank the anonymous reviewers for their helpful comments and suggestions for improving the manuscript.

REFERENCES

[1] Zheng, X. S.—Pulli, P.: Improving Mobile Services Design: A QFD Approach. Computing and Informatics, Vol. 26, 2007, pp. 369–381.

[2] Chakraborty, D.—Joshi, A.—Yesha, Y.—Finin, T.: Towards Distributed Service Discovery in Pervasive Computing Environments. IEEE Transactions on Mobile Computing, Vol. 5, 2006, No. 2, pp. 97–112.

[3] Sharmin, M.—Ahmed, S.—Ahamed, S. I.: An Adaptive Lightweight Trust Reliant Secure Resource Discovery for Pervasive Computing Environments. Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM 2006), Pisa, Italy, 2006, pp. 258–263.

[4] Gao, Z. G.—Yang, Y. T.—Zhao, J.—Cui, J. W.—Li, X.: Service Discovery Protocols for MANETs: A Survey. Proceedings of the Second International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2006), Hong Kong, China, 2006, pp. 232–243.

[5] Arnold, K.—Osullivan, B.—Scheifler, R. W.—Waldo, J. et al.: The Jini Specification (The Jini Technology). Reading, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, June 1999.

[6] John, R.: UPnP, Jini and Salutaion – A Look at Some Popular Coordination Frameworks for Future Network Devices. Technical Report, California Software Labs, 1999, Available on: `http://www.calsoftlabs.com/whitepapers/upnp-devices.html`.

[7] OASIS UDDI Specification Technical Committee: Universal Description Discovery and Integration Platform. July 2002, Available on: `http://www.oasis-open.org/specs/index.php\#uddiv3.0.2`.

[8] Guttman, E.—Perkins, C.—Veizades, J.—Day, M.: Service Location Protocol. Version 2. IETF RFC 2608, June 1999, available on: `http://www.faqs.org/rfcs/rfc2608.html`.

[9] Choonhwa, L.—Helal, A.—Desai, N.—Verma, V.—Arslan, B.: Konark: A System and Protocols for Device Independent. Peer-to-Peer Discovery and Delivery of Mobile Services. IEEE Transactions on Systems, Man and Cybernetics, Part A, Vol. 33, 2003, No. 6, pp. 682–696.

[10] Klein, M.—Konig-Ries, B.—Obreiter, P.: Lanes–A Lightweight Overlay for Service Discovery in Mobile Ad Hoc Networks. Proceedings of the 3rd Workshop on Applications and Services in Wireless Networks (ASWN 2003), Berne, Switzerland, July 2–4, 2003, pp. 101–112.

[11] Klein, M.—Konig-Ries, B.—Obreiter, P.: Service Rings – A Semantic Overlay for Service Discovery in Ad Hoc Networks. Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA2003), Washington, DC, USA, Sep. 1–5, 2003, pp. 180–185.

[12] Chakraborty, D.—Joshi, A.—Yesha, Y.—Finin, T.: GSD: A Novel Group-based Service Discovery Protocol for MANETs. Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002), Stockholm, Sweden, 2002, pp. 140–144.

[13] Gao, Z. G.—Wang, L.—Yang, X. Z.—Wen, D. X.: PCPGSD: An Enhanced GSD Service Discovery Protocol for MANETs. Computer Communications, Vol. 29, 2006, No. 12, pp. 2433–2445.

[14] Gao, Z. G.—Wang, L.—Yang, M.—Yang, X. Z.: CNPGSDP: An Efficient Group-based Service Discovery Protocol for MANETs. Computer Networks, Vol. 50, 2006, No. 16, pp. 3165–3182.

[15] Gao, Z. G.—Yang, X. Z.—Cai, S. B.: Flexible Forward Probability Based Service Discovery Protocol for MANETs. Journal of Harbin Institute of Technology, Vol. 37, 2005, No. 9, pp. 1256–1260 (in Chinese).

[16] Wireless Adaptive Mobility Lab. DEPT of Comp. SCI, UCLA: Glomosim: A Scalable Simulation Environment for Wireless and Wired Network System. Available on: `http://pcl.cs.ucla.edu/projects/glomosim/`.

Qianli Ma received her B. Sc. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1990, and then received her M. Sc. degree in industrial automation from University of Science and Technology Beijing, Beijing, China, in 1997. In 2009, she received her Ph. D. degree in computer science and technology at Harbin Institute of Technology, Harbin, China. She is currently an Associate Professor in School of Computer and Information Technology at Shanxi University, Taiyuan, China. Her research interests include service discovery, context-aware and pervasive computing.

Minghong Liao is a Professor in School of Computer Science and Technology of Harbin Institute of Technology. His current research interests include new operating system, real-time and embedded computing, service discovery, peer-to-peer computing and artificial intelligence.

Shouxu Jiang is a Professor in School of Computer Science and Technology of Harbin Institute of Technology. His current research interests include service discovery, peer-to-peer computing and wireless sensor networks.

Zhenguo Gao received his B. Sc. and M. Sc. degree in mechanical and electrical engineering from Harbin Institute of Technology, Harbin, China, in 1999 and 2001, respectively, and then received his Ph. D. degree in computer architecture from Harbin Institute of Technology in 2006. His research interests include service discovery, broadcasting and routing for cognitive radio at ad hoc networks.

Wenxu Xu received his BS and MS degree in computer science from Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively. He is currently a Ph. D. candidate in computer science and technology at Harbin Institute of Technology, Harbin, China. His research interests include data mining, peer-to-peer computing and quantum computation.