A Generic Minimum Dominating Forward Node Set Based Service Discovery Protocol for MANETs

Zhenguo Gao[1], Sheng Liu[1], Mei Yang[2], Jinhua Zhao[3], and Jiguang Song[1]

[1] College of Automation, Harbin Engineering University, Harbin, 150001, China
{gag,liusheng,songjg}@hrbeu.edu.cn
[2] Department of Electrical and Computer Engineering University of Nevada, Las Vegas, NV 89154, USA meiyang@egr.unlv.edu
[3] School of Astronautics, Harbin Institute of Technology, Harbin, 150001, China
zhaojinhua@hit.edu.cn

Abstract. Service discovery is a crucial feature for the usability of mobile ad-hoc networks (MANETs). In this paper, Minimum Dominating Forward Node Set based Service Discovery Protocol (MDFNSSDP) is proposed. MDFNSSDP has the following characteristics: 1)It minimizes the number of coverage demanding nodes in the current node's 2-hop neighbor set. 2) It minimizes the size of dominating forward node set used to cover all coverage demanding nodes. Forward nodes are selected based on local topology information and history information piggybacked in service request packets. Only these forward nodes are responsible for forwarding service request packets. 3) The coverage of service request packets is guaranteed. 4) Multiple service requests can be fulfilled in just one service discovery session. Simulations show that MDFNSSDP is an effective, efficient, and prompt service discovery protocol for MANETs.[1]

1 Introduction

Mobile Ad-Hoc Networks (MANETs)[1] are temporary infrastructure-less multihop 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 features that can be utilized or benefited by any node; Service description of a service is the information that describes the service's characteristics, such as types and

[1] This work is supported by National Postdoctoral Research Program (No.NPD060234), Postdoctoral Research Program of HeiLongJiang (No.HLJPD060234), Fundamental Research Foundation of Harbin Engineering University (No.HEUFT06009), Fostering Fundamental Research Foundation of Harbin Engineering University (No.HEUFP07001).

R. Perrott et al. (Eds.): HPCC 2007, LNCS 4782, pp. 634–646, 2007.

attributes, access methods, 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 request packets, 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 session. Coverage preservability of a service discovery protocol is the property that the coverage of service discovery requests when the protocol is used is the same to that when flood policy is used.

The objective of service discovery protocol is to reduce service request packets redundancy while retaining service discoverability. Usually this is approached by searching for more efficient policy of service request packet forwarding.

Existing service discovery protocols targeted at MANETs are not very applicable for their variety of problems. Flood-based protocols (Konark[3], Proximity SDP[4]) face the risk of broadcast storm problem[5]. Some other flood-like protocols (FFPSDP[6], RICFFP[7]) can not guarantee the coverage of service request packets, which leads to lower service discovery ratio. Two group-based protocols (GSD[8], Allia[9]) cause serious redundancy of unicast request packets. 2-layer protocols[10][11] construct upper logic layer by selecting out some nodes with variety of criterions. Upper logic layer requires costly maintenance. DSDP[12] is superior to other 2-layer protocols for its lower topology maintenance overhead. In multi layer protocols (ServiceRing[13], MultiLayerSDP[14]), their hierarchy architectures are hard to maintain, which has been verified through simulations[15]. Additionally, no existing protocols consider the requirement of fulfilling multiple service discovery requests in one service discovery session. For a more detailed survey please ref to ref[16].

Noticing above problems and the importance of coverage preservability property of service discovery protocols, Minimum Dominating Forward Node Set based Service Discovery Protocol (MDFNSSDP) is proposed in this paper. The new scheme minimizes the number of coverage demanding nodes in the current node's 2-hop neighbor set, and finds a minimum dominating forward node set to cover all these coverage demanding nodes. In this way, service request packet overhead is greatly reduced. MDFNSSDP preserves the coverage of service discovery sessions, and it can fulfill multiple service discovery requests in just one service discovery session. Besides, it offers some user-definable parameters, which can be used to adjust its operation.

Dominating set scheme has been applied to many aspects in MANET, including routing, broadcasting, and applications, but no similar work is found used in service discovery protocol. Besides, most works are either omitted the necessity to minimize the set to be covered or lack of coverage preservability proof. Some work using dominating set can be found in ref[17].

The rest of the paper is organized as follows. In Section 2, data structures of MDFNSSDP and some definitions are given. A problem called dominating forward node set (DFNS) problem is proposed and its NP-completeness is proved. An heuristic to solve the DFNS problem is proposed, and its approximation ratio

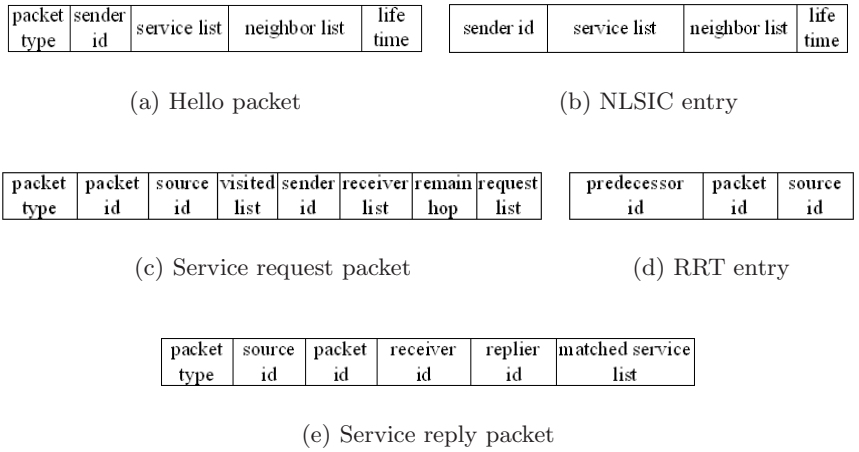
is analyzed. In Section 4, operations of MDFNSSDP are described and demonstrated. In Section 5, some properties of MDFNSSDP, such as the existence of dominating forward node set and coverage preservability, are proved. In Section 6, the performance of MDFNSSDP is analyzed through extensive simulations. Finally, in Section 7, a conclusion is made.

2 Preliminaries of MDFNSSDP

2.1 Data Structures

In MDFNSSDP, each node broadcasts hello packets periodically. The structure of hello packets is shown in Fig. 1(a). The *packet-type* field indicates the type of the packet. The *sender-id* field indicates the sender of the packet. The *service-list* field stores the list of local services' descriptions. The *life-time* field indicates the time that the packet can be cached by others. The *neighbor-list* field stores the list of the current node's 1-hop neighbors.

Hello packets received will be cached in NLSIC for a period. The structure of NLSIC item is shown in Fig. 1(b).



(a) Hello packet

(b) NLSIC entry

(c) Service request packet

(d) RRT entry

(e) Service reply packet

Fig. 1. Data structures in MDFNSSDP protocol

The structure of service request packets is shown in Fig. 1(c). The *packet-id* filed is a number that increases monotonically with each service request packet generated by a node. This field identifies different service request packets from the same node. The *source-id* filed indicates the client that generates the service request packet. The *visited-list* filed stores the list of nodes that the packet has passed. Depends on the value of protocol parameter, the *visited-list* field may also store the list of 1-hop or 2-hop neighbors. The *sender-id* filed indicates the direct sender of the packet. The *receiver-list* filed stores the list of so called forward nodes that are responsible for forwarding the request packet further. The

remain-hop filed indicates the number of hops that the packet can still travel. The *request-list* field describes the list of the descriptions of the requested services as well as whether any matched services has been found for each requested service.

Each node maintains a RRT which is used in two tasks: 1) check for duplicated request packets, and 2) forward service reply packets reversely to the corresponding client node. Fig. 1(d) shows the structure of an RRT entry. The predecessor-id field indicates the node from which the service request packet is received. It is just the next hop node that a corresponding reply packet should be forwarded to. The *packet-id* and *source-id* field are the same to that of the service request packet

The structure of service reply packets is shown in Fig. 1(e). The *packet-type* field indicates the type of the packet. The *source-id* field stores the destination of the reply packet, which is just the node that generates the corresponding request packet. The *packet-id* field stores the packet-id of the corresponding request packet. The *receiver-id* field indicates the next-hop node of the service reply packet. The *replier-id* field indicates the node that generates the reply packet. The *matched-service-list* field stores the description of the matched services.

2.2 Notations and Definitions

The following notations are used in the rest of the paper.

u: the current node.
s: the client that generates the service request packet.
v: the direct sender of the current service request packet.
$N_x(u)$: the set of nodes that are at most x-hop away from node u, i.e., node u's x-hop neighbor set, including u itself.
$H_x(u)$: the set of nodes that are just x-hop away from node u.
$V(v)$ the set of nodes in the *visited-list* field of the request packet sent by v.
$F(v,u)$: $F(v,u) = H_1(u) - N_1(v)$.
$R(u)$: The set of nodes in the *receiver-list* field of request packet sent by u.

We make the following definitions.

Def. 1 (Forward Node): Each node in $R(u)$ is a forward node of node u.

Def. 2 (Candidate Forward Node): Each node in $F(v,u)$ is a candidate forward node of node u. Obviously, $R(u) \subseteq F(v,u)$ since that forward nodes are all selected from $F(v,u)$.

Def. 3 (Coverage of a service discovery session): All nodes that could receive or have received service request packets of a service discovery session are called as in the coverage of the service discovery session. It can also be said that these nodes are covered by the service discovery session.

Def. 4 (Extendedly Covered): For a node w, if $\exists x \in R(u)$ meets $w \in H_1(x)$, then we say that node w is extendedly covered by node u.

Def. 5 (Coverage Demanding Node): To obtain coverage preservability, when forwarding service request packets, all node in $H_2(u)$ should be guaranteed

to be covered by the current service discovery session. Some nodes in $H_2(u)$ have already been covered or to be covered through other ways. Hence, only a subset of $H_2(u)$ should be guaranteed to be extendedly covered by the current node. These nodes are called as coverage demanding nodes. The set of coverage demanding nodes is denoted as $H_{CDN}(v,u)$. Obviously, $H_{CDN}(v,u) \subseteq H2(u)$. In MDFNSSDP, $H_{CDN}(v,u) = H_2(u) - N_2(V(v))$.

Def. 6 (Dominating Forward Node Set): Given a set of nodes $R(u)$, if for $\forall w \in H_{CDN}(v,u)$, $\exists x \in R(u)$ that meets $w \in H_1(x)$, or in other words , $H_{CDN}(v,u) \subseteq H_1(R(u))$, then $R(u)$ is a dominating forward node set, denoted as $F_{DFNS}(v,u)$.

3 Find a Minimum Dominating Forward Node Set

When receiving a service request packet, each forward node will have to forward the packet further, unless all requested services are found or the packet's hop-limit is reached. Hence, in order to reduce request packet overhead, the size of DFNS should be minimized. The task of finding a DFNS with minimum size is called as DFNS problem.

Using H to represent the set of coverage-needed hidden servers $H_{CDN}(v,u)$, C to represent the family of sets $\{H_1(w)|w \in F(v,u)\}$, and F to represent $F(v,u)$, the DFNS problem can be defined formally as follows:

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

Since that $\bigcup_{w \in F(v,u)} H_1(w) \supseteq H_2u - N_2V(v) = H_{CDN}(v,u)$(refer to Lemma 3 in the following text), and there is a 1-to-1 correspondence between C and F, the decision version DFNS Problem can be defined formally as follows:

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

Lemma 1. *The decision version of the DFNS problem is NP-complete.*

Proof. The proof is omitted for the limit of space. □

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

Heuristic: greedy Minimum DFNS heuristic.

1. Let $F_{DFNS}(v,u) = \Phi$(empty set), $H_{RC}(v,u) = \Phi(H_{RC}(v,u)$ is a temporal set used to store node set already covered by $F_{DFNS}(v,u))$.
2. Find node $w \in F(v,u)$ with maximum $|H_1(w) \bigcap H_{CDN}(v,u) - H_{RC}(v,u)|$. In case of a new tie, select w with smallest ID.
3. $F_{DFNS}(v,u) = F_{DFNS}(v,u) + w, F(v,u) = F(v,u) - w$.
4. $H_{RC}(v,u) = H_{RC}(v,u) \bigcup H_1(w)$. If $H_{RC}(v,u) \supseteq H_{CDN}(v,u)$, exit. Otherwise, go to step 2.

Lemma 2. *The approximation ratio of the greedy minimum DFNS heuristic is* $ln(max_{z \in F(v,u)} |H_1(z) \bigcap H_{CDN}(v,u)|)$.

Proof. The proof is omitted for the limit of space. □

4 The Operations of MDFNSSDP Protocol

MDFNSSDP protocol has three basic operations: Hello packet exchanging, service request packet forwarding, and service reply packet routing.

4.1 Hello Packet Exchanging

In MDFNSSDP, each node in a MANET should broadcast hello packets periodically. Node can cache the information in the hello packets into their NLSIC. This information will be kept valid for a user-defined period. Hello packets can only travel 1 hop, i.e., nodes should not forward a reply packet further. New hello packets are constructed basing the information in NLSIC.

The content of the *service-list* field of a hello packet is determined by protocol parameter S_{TYPE} as follows.

- If S_{TYPE}=NONE, this field contains nothing.
- If S_{TYPE}=SELF, this field contains the descriptions of all services provided by the node.
- If S_{TYPE}=HOP1, this field contains not only the descriptions of the services provided by the node, but also those of the services provided by the current node's neighbors.
- If S_{TYPE}=HOP2, the field contains the descriptions of the services of three sources: 1)the current node, 2)1-hop neighbors of the current node, and 3) 2-hop neighbors of the current node.

4.2 Service Request Packet Forwarding

When needing services, a node firstly checks to see if there are any matched services for each requested service, either provided by the node itself or found from its NLSIC. If yes, the service discovery request succeeds. If no, the node constructs a service request packet and sent it out. The *request-list* field of a request packet can contain multiple service requests. Hence, in MDFNSSDP, multiple service discovery requests can be fulfilled in one service discovery session. The maximum number of service discovery requests in *request-list* field is determined by parameter CSize.

Request Packet Forward Decision. When receiving a service request packet from other nodes, or the current node needs services, the node should determine whether to forward the packet or not. If the following 4 conditions are all met, the packet should be forwarded. The 4th condition is important to the property of fulfilling multiple service discovery requests in just one service discovery session, called as multi task mode.

- This is a new service request packet for the node. Whether a service request packet is new or not for the receiver can be determined basing on the current node's RRT.
- The *remain-hop* field of the service request packet is bigger than 0.
- The current node is in the list of forward nodes in the *receiver-list* field of the service request packet.
- There are some unmatched service requests yet.

Request Packet Forwarding Procedure. Following 5 steps are used to forward a service request packet.

1. Determine the set of coverage demanding nodes, $H_{CDN}(v,u) = H_2(u) - N_2(V(v))$
2. Find a minimum dominating forward node set. Using the previous greedy-based DFNS heuristic to find a dominating set $F_{DFNS}(v,u)$ from $H_{CDN}(v,u), F(v,u)$ and $H_1(w)|w \in H_1(u)$.
3. Enclose nodes in $F_{DFNS}(v,u)$ into the the packet's *receiver-list* field.
4. Update the contents of other fields of the request packet (section 3.2.3).
5. Send the service request packet out in broadcast mode.

Request Packet Content Update. Before forwarding a service request packet, some fields of the service request packet should be updated as follows. The content of the *visited-list* field depends on the parameter V_{TYPE}, whereas the size of the items in *visited-list* field is determined by V_{SIZE}.

- If V_{TYPE} =NONE, *visited-list*field contains nothing. If V_{TYPE}=SELF, *visited-list* field contains the identity of the current node.If V_{TYPE}=HOP1, *visited-list* field contains $N_1(u) = u + H_1(u)$.If V_{TYPE}=HOP2, *visited-list* field contains $N_2(u) = u + H_1(u) + H_2(u)$.
- If the number of items in the *visited-list* field is bigger than V_{SIZE}¿0, the earliest item is removed from the *visited-list* field to make room for the latest item. If V_{SIZE}=-1, the number of items is not limited.
- In *request-list* field, all matched service requests are designated as "matched".
- The *remain-hop* field is decreased by 1.

4.3 Service Reply Packet Routing

When receiving a service request packet, each node that finds matched services should construct a service reply packet and send it out, no matter what the source of the matched services are, the node itself or its NLSIC.

When receiving a service reply packet, a node firstly checks if this node is just the destination of the packet. If yes, the node caches the service information in reply packet, and this service discovery session finishes. If no, the node then checks if there are any matched services not seen in previous reply packets. If yes, the service reply packet will be forwarded further. Otherwise it will be discarded.

To forward a service reply packet, the node searches for the RRT item that corresponds to the service reply packet. Then the service reply packet is forwarded to the node indicated by the predecessor-id field of the founded item. In this way, a service reply packet will be relayed to the source of the corresponding service request packet along the reverse path.

5 Property Analysis of MDFNSSDP

5.1 Existence of Dominating Forward Node Set

In MDFNSSDP, $H_{CDN}(v,u)$ is determined by removing $N_2(V(v))$ from $H_2(u)$, and then a set $F_{DFNS}(v,u)$ is selected from $F(v,u)$ to cover $H_{CDN}(v,u)$.

This section focuses on two aspects: 1) the possibility of finding a qualified $F_{DFNS}(v,u)$, 2) the property of coverage preservability of MDFNSSDP.

Lemma 3.*A qualified $F_{DNFS}(v,u)$ can be found in $F(v,u)$ to cover $H_{CDN}(v,u)$.*

Proof. The proof is omitted for limited space. □

5.2 Coverage Preservability of MDFNSSDP

In MDFNSSDP, the set of coverage demanding nodes, $H_{CDN}(v,u)$, is determined as $H_2(u) - N_2(V(v))$. This configuration can still guarantee the coverage of service discovery sessions. This is to say, the coverage of service discovery requests when MDFNSSDP is used is the same to that when flood policy is used. About the correctness of this property, there are the following lemmas. All these lemmas are based on the following assumptions:

Assumption 1: The underlining MAC protocol is ideal. That is to say, each packet sent by a node will be received by its neighbors correctly and timely, and there are no collisions.

Assumption 2: The value of the remain-hop field of a request packet is large enough to cover the network, i.e., the restriction of remain-hop is not considered.

Assumption 3: The effect of cancellation of service request packet forwarding by a node where all requested services have been matched is not considered.

Lemma 4. *In MDFNSSDP, if a node has received a service request packet of a service discovery session, then all neighbors of the node must be able to receive a service request packet of the service discovery session.*

Proof. The proof is omitted for limited space. □

Lemma 5. *In MDFNSSDP, if a MANET is connected, then all nodes in the MANET must be able to receive a service request packet of a service discovery session.*

Proof. In a connected MANET, for each node pair, there are paths of 1 or more hops. Suppose the client of a service discovery session is node s, then for each node w in the MANET, there must be a path. Suppose $s \to x_1 \to x_2 \to \cdots \to x_n \to w$ is a path from node s to w, now we prove that node w must be able to receive a service request packet of the service discovery session.

According to Assumption 1, $x_1 \in H_1(s)$ must be able to receive the service request packet sent by node s. Then using Lemma 4 repeatedly along the path $s \to x_1 \to x_2 \to \cdots \to x_n \to w$, nodes $x_1, x_2, \cdots, x_n, w$ must also be able to receive request packets of the service discovery session. Hence, the lemma follows. □

6 Simulation Analysis

6.1 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 reflects the efficiency the policy of forwarding service request packets.
- **Succeeded-SDP-Number:** It is the number of service discovery sessions in which the client has received at least one successful 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 service discovery sessions. It measures the promptness of service discovery protocols. It also reflects the average distance between clients and the corresponding first repliers.
- **Ratio of Succeeded-SDP-Number to Total-SDP-packet-number (Suc2Total)**: This metric is the ratio of Succeeded-SDP-number to the sum of service request packets and service reply packets. It reflects the efficiency of service discovery protocols. The number of periodical packets, such as hello packets in MDFNSSDP, is sensitive to protocol parameters, and almost all service discovery protocols generates such packets. Hence, to make a more discriminative inspection on protocol performance, periodical packets are not calculated in this metric.

6.2 Simulation Models

Simulation studies are performed using Glomosim[19]. The distributed coordination function (DCF) of IEEE 802.11 is used as the underlying MAC protocol. Random Waypoint Model (RWM) is used as the mobility model.

In RWM mobility model, nodes move towards their destinations with a randomly selected speed $V \in [V_{min}, V_{max}]$. When reaching its destination, a node keeps static for a random period $T_P \in [T_{min}, T_{max}]$. When the period expires, the node randomly selects a new destination and a new speed, then moves to the new destination with the new speed. The process repeats permanently. In our simulations, $T_{min} = T_{max} = 0, V_{min} = V_{max} = V$.

6.3 Select Comparative Solutions

To make a comparative study, we implement MDFNSSDP and other three typical service discovery protocols for MANETs: BASIC, GSD[8], SSDP[2].

BASIC represents the most straightforward service discovery protocol where each node should forward each service request packet it received, unless the packet reaches its hop limit. BASIC method is widely accepted as a benchmark in evaluating service discovery protocols.

In GSD, services are classified into groups. Each server generates service advertisement packets periodically. Through advertisement packets, each node knows the services provided by its adjacent nodes as well as the group information of some services that these adjacent nodes have seen in received service advertisement packets. When forwarding a service request packet, a node can intelligently forward the packet towards some selected nodes in unicast mode. These selected nodes have seen some services of the same group as the requested service.

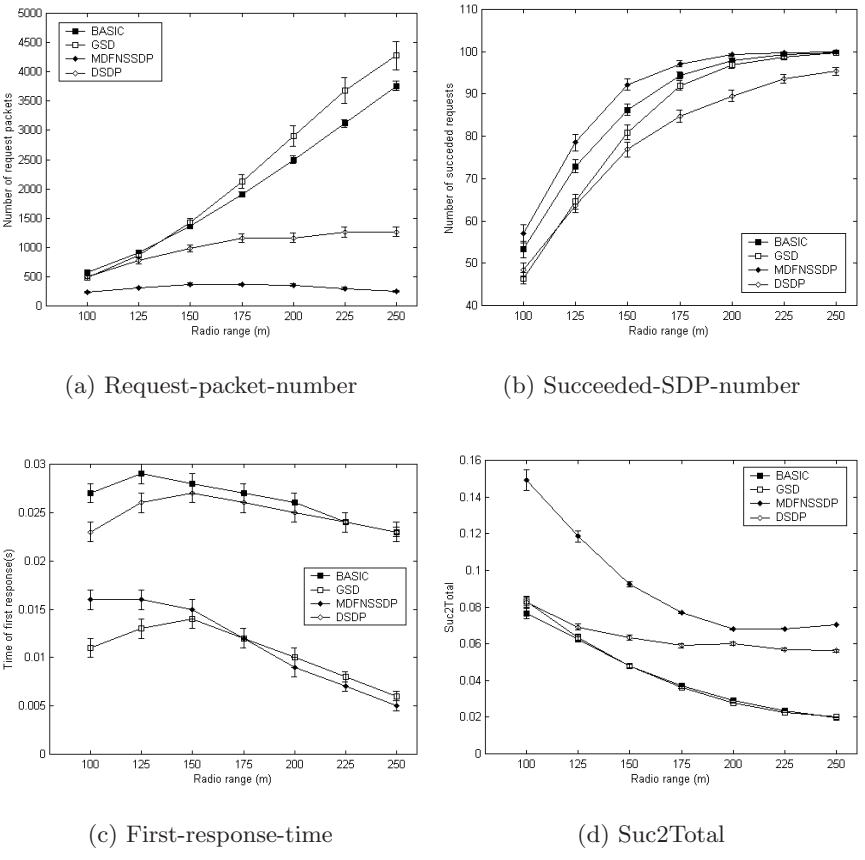
In DSDP, some nodes are selected out according to node degree and link stability to construct an upper layer logic backbone. DSDP works in three stages: hello packet collection, backbone node selection, backbone maintenance. They are all based on periodical hello packets. After the construction of backbone, all service discovery packets will spread along the backbone, and thus service request packets will be reduced.

6.4 Simulation Results

Seven simulaiton sets were performed with radio range set to 100m, 125m, 150m, 175m, 200m, 225m, 250m, respectively. Each simulation set contains 4 subsets, which adopts the four protocols, especially. Each subset includes 100 similar simulations with different random seeds. Simulation scenarios are created with 100 nodes initially distributed according to the steady state distribution of random waypoint mobility model. At the beginning of each simulation, 100 nodes are randomly distributed in the scenario area, and predetermined number of nodes are randomly selected as servers. These selected servers provide randomly selected services. During each simulation, 100 service discovery sessions are started at

Table 1. Basic parameters in simulation study

Parameters	Value	Parameters	Value
Scenario	1000m×1000m	Number of service groups	2
Number of nodes	100	Number of services in each group	5
Simulation time	1000s	Hop of broadcast packets(GSD)	1
bandwidth	1Mbps	Hello(broadcast) packet interval	20s
Session number	100	Valid time of cache item	30s
SType(MDFNSSDP)	SELF	Node speed(m/s)	0m/s
VType(MDFNSSDP)	SELF	Hop of request packets	3
VSize(MDFNSSDP)	-1	Number of servers	100
CSize	1		



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

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

Fig. 2. Protocol performance under different radio range

randomly selected time by randomly selected nodes.Some basic simulation parameters are listed in Table 1, and simulation results are shown in Fig. 5 with error bars report 95% confidence.

Fig.2 shows that under different radio range, MDFNSSDP has the least request packet overhead (Fig.2(a)), the highest service discoverability (Fig. 5(b)), the quickest response (Fig.2(c)), and the highest efficiency (Fig.2(d)).

7 Conclusions

In this paper, Minimum Dominating Forward Node Set based Service Discovery Protocol (MDFNSSDP) for MANETs is proposed. MDFNSSDP protocol has the following properties:

– MDFNSSDP can fulfill multiple service discovery requests in just one service discovery session. The request-list field can store multiple service requests, and protocol operations are tailored elaborately for multiple-request tasks.

- MDFNSSDP preserves the coverage of service discovery sessions.
- MDFNSSDP minimizes the size of coverage demanding node set $H_{CDN}(v, u)$.
- MDFNSSDP minimizes the number of selected forward nodes by finding a minimum dominating forward node set to cover all coverage demanding nodes using a greedy-based DFNS heuristic.

Simulation results show that MDFNSSDP is an efficient, effective, and prompt service discovery protocol for MANETs.

References

1. IETF, Mobile ad-hoc network (MANET) working group. [Online]. Available: `http://www.ietf.org/html.charters/manet-charter.html`
2. Kozat, U.C., Tassiulas, L.: Service discovery in mobile ad hoc networks: an overall perspective on architecture choices and network layer support issues. Ad Hoc Networks, 23–44 (2004)
3. Helal, S., Desai, N., Verma, V., Lee, C.: Konark - a service discovery and delivery protocol for ad-hoc networks. In: WCNC 2003. New Orleans, USA pp. 2107–2133 (2003)
4. Azondekon, V., Barbeau, M., Liscano, R.: Service selection in networks based on proximity confirmation using infrared. In: ICT 2002. Beijing, China, pp. 116–120 (2002)
5. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. ACM Wireless Networks, 153–167 (2002)
6. Gao, Z.G., Yang, X.Z., Cai, S.: FFPSDP: flexible forward probability based service discovery protocol. Journal of Harbin Institute of Technology 1265–1270 (2005)
7. Gao, Z.G., Yang, X.Z., Ma, T.Y., Cai, S.B.: RICFFP an efficient service discovery protocol for MANETs. In: Yang, L.T., Guo, M., Gao, G.R., Jha, N.K. (eds.) EUC 2004. LNCS, vol. 3207, pp. 786–795. Springer, Heidelberg (2004)
8. Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: GSD: a Novel Group-based Service Discovery Protocol for MANETs. In: MWCN 2002. Stockholm, Sweden, pp. 140–144 (2002)
9. Ratsimor, O., Chakarborty, D., Joshi, A., Finin, T.: Allia: alliance-based service discovery for ad-hoc environments. In: Păun, G., Rozenberg, G., Salomaa, A., Zandron, C. (eds.) Membrane Computing. LNCS, vol. 2597, pp. 1–9. Springer, Heidelberg (2003)
10. Nordbotten, N.A., Skeie, T., Aakvaag, N.D.: Methods for service discovery in bluetooth scatternets. Computer Communications, 1087–1096 (2004)
11. Liu, J.C., Zhang, Q., Zhu, W.W., Li, B.: Service locating for large-scale mobile ad hoc network. International Journal of Wireless Information Networks, 33–40 (2003)
12. Yoon, H.J., Lee, E.J., Jeong, H., Kim, J.S.: Proximity-based overlay routing for service discovery in mobile ad hoc networks. In: Aykanat, C., Dayar, T., Körpeoğlu, İ. (eds.) ISCIS 2004. LNCS, vol. 3280, pp. 176–186. Springer, Heidelberg (2004)
13. Klein, M., Ries, B.K., Obreiter, P.: Service rings - a semantic overlay for service discovery in ad hoc networks. In: Mařík, V., Štěpánková, O., Retschitzegger, W. (eds.) DEXA 2003. LNCS, vol. 2736, pp. 180–185. Springer, Heidelberg (2003)
14. Klein, M., Ries, B.K.: Multi-layer clusters in ad-hoc networks - an approach to service discovery. In: IWP2PC'02. Pisa, Italy, pp. 187–201 (2002)

15. Klein, M., Hoffman, M., Matheis, D. et al.: Comparison of overlay mechanims for service trading in ad hoc networks. TR. 2004-2, University of Karlsruhe (2004)
16. Gao, Z.G., Yang, Y.T., Zhao, J., Cui, J.W., Li, X.: Service discovery protocols for MANETs: a survey. In: Cao, J., Stojmenovic, I., Jia, X., Das, S.K. (eds.) MSN 2006. LNCS, vol. 4325, pp. 232–243. Springer, Heidelberg (2006)
17. Brad, W., Tracy, C.: Comparison of broadcasting techniques for mobile ad hoc networks. In: MobiHoc 2002, Lausanne, Switzerland, pp. 194–205 (2002)
18. Chvatal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research, 233–235 (1979)
19. Glomosim: a scalable simulation environment for wireless and wired network system. Available: `http://pcl.cs.ucla.edu/projects/domains/glomosim.html`