

Adaptive Neighbor Selection for Service Discovery in Mobile Ad Hoc Networks*

Eunyoung Kang¹, Yongsoon Im², and Ungmo Kim¹

¹ School of Computer Engineering, Sungkyunkwan University,
440-776, Suwon, Gyeonggi-do, Korea
{eykang, umkim}@ece.skku.ac.kr

² School of Broadcasting, Kookje College,
459-070, Pyeongtaek, Gyeonggi-do, Korea
ysim@kookje.ac.kr

Abstract. Service discovery to search for an available service in a mobile ad-hoc network is an important issue. Although mobile computing technologies grow ever more powerful and accessible, MANET, consisting of mobile devices without any fixed infrastructure, has such features as high mobility and resource constraints. Given these features, the costs of service discovery in mobile ad-hoc networks must be lower than those of service discovery in conventional fixed networks. In this paper, we design and evaluate a service discovery scheme to effectively discover services by using only local information in a mobile ad hoc network. Our service discovery protocol is based on the concept of peer-to-peer caching of service advertisement and node ID-based forwarding of service requests to solve these problems. Our protocol is that physical hop counts and the number of messages exchanged have been significantly reduced, since it does not require a central lookup server and does not rely on multicasting and flooding. The simulation results show that, in the proposed scheme, physical hop counts and the number of messages exchanged have been significantly reduced, compared with the other protocol.

Keywords: Service discovery, Peer-to-Peer Caching, MANET, Message Delivery.

1 Introduction

A mobile ad-hoc network (MANET) autonomously composed of mobile nodes independent of the existing wired networks or base stations has recently attracted substantial interest from industrial or research groups. Because it lacks infrastructure support,

* This work was supported in part by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement, IITA-2008-C1090-0801-0028) and by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy(MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-B1-10M.

each node acts as a router, forwarding messages for other nodes [1], [2]. In accordance with these trends, such mobile devices as PDAs, handheld devices and notebook computers have rapidly evolved. Due to the development of those mobile devices increasing user demand, file sharing or service discovery is emerging as an important issue.

With regard to file sharing or service discovery, there have been lots of researches on wired network P2P systems. Examples of representative unstructured P2P systems include Gnutella [3] and KaZaA [4]. Because Gnutella uses centralized directory services, it can easily find the location of each directory. But it causes a central directory server to produce a number of query messages, incurring bottlenecks. On the other hand, such structured P2P systems as Chord [5], CAN [6], and Pastry [7] are based on distributed hash tables (DHTs). A DHT-based overlay network makes out a table for searching by using a file and a key value obtained as a result of applying the file to its hash function. It is effective for peers involving themselves in a network to distribute and save the whole file information for searching because such searching is based on the key value corresponding to the file.

Notwithstanding advantages as demonstrated above, there is little application of these technologies to a mobile ad-hoc network. It is difficult to apply P2P applications based on a fixed network environment to ad-hoc networks different from those fixed wired networks, because their nodes are free to move. Although mobile computing technologies grow ever more powerful and accessible, mobile devices have a lower level of processing capacity and use batteries of limited power. They consume substantial power when they exchange messages. In this sense, it is requirement that costs are reduced for P2P applications with mobile devices in a wireless network. An ad-hoc network enables reduction of energy consumption, and reduction of wireless bandwidth needed for transmission, by reducing query messages among P2P applications in the network.

To solve these problems, this study proposes an improved service discovery protocol which is based on the concept of peer-to-peer caching of service advertisement and the lower node ID-based forwarding of service requests. First, it listen service information from neighbor node and store service information in own local service cache for the purpose of caching service advertisement. Second, it utilizes the lower node ID scheme to efficiently discover service information in MANET. Regarding lower ID delivery, a node with lower ID receives information from another node since messages are transmitted to the node of lower ID. The node of the lowest ID operates as a distributed index server. The number of messages exchanged in a network and average hop counts between a service requester and a service provider are reduced since the probability of responding service messages increases. The results of simulations showed that the proposed scheme reduced total network cost and power consumption by reducing messages and average hop counts(query delay), and improving response time, by comparison with any other system.

The remainder of this paper is organized as follows: in Section 2, the existing various approaches for service discovery are covered; Section 3 gives an explanation to the proposed service discovery architecture; an analysis of the architecture and its validity are exploited in Section 4; Finally, Section 5 concludes this paper, proposing the future study.

2 Existing Approaches for Service Discovery

Service discovery is an important area of research in wired networks and as well as wireless networks. There has been studied in the field of wired networks to develop peer-to-peer architectures as shown in [3-7]. To provide service discovery, the existing works use P2Ps including Gnutella [3] and KaZaA [4]. The former mainly uses central directory services. That is, each server identifies their location at which it registers its own services so that clients find services by sending service request queries to the corresponding directory where their own services are registered. On the other hand, service discovery protocols such as KaZaA generally use flooding search protocol, creating lots of query messages. These kinds of networks have to be controlled by a central server, leading to the problem of creating a large number of query messages. Structural P2P networks, such as Chord [5], CAN [6], and Pastry [7], are based on DHT in which each service is allocated ID in accordance with available keys. Due to the dynamic characteristics such as node mobility, resource (CPU, bandwidth, storage) constraints, and limited transmission range, using it on an ad-hoc network is not valid solution.

Jini [8], SLP2 [9], Salutation [10], Konark [11], Allia [12], and UPnP [13] are service discovery architectures of an ad-hoc network. These researches have been developed over the past few years to efficiently discover wired infrastructure-based services from wired and as well as wireless platforms. Jini, UPnP and SLP2 are service discovery protocols for a one-hop ad-hoc network environment. They are not suitable for an environment with high mobility, where lots of nodes participate in a network. In addition, Jini and SLP2 provide centralized service information management. Jini registers the service objects of a server through lookup services, and SLP2 manages the service information of a server through directory agents. On the other hand, UPnP, Salutation, Konark, and Allia collect and manage information at their own cache through service advertisement based on a P2P system. Regarding Konark, multicast groups forwards messages. In Allia, various alliances among nodes are formed to forward messages to allied nodes. GSD [14] is based on group-based intelligent forwarding of service requested. The group means a service group to which the requested service belongs. They selectively forwards the request to neighboring nodes belongs to one of those groups. However, a node that has high degree of node connectivity often broadcast to neighboring nodes.

3 Proposed Service Discovery Architecture

3.1 Single Mobile Device Architecture

Fig. 1 shows components included in a single mobile device and the proposed service discovery architecture. The service discovery architecture is composed of 3 layers: application layer, service management layer and network communication layer.

Application layer provides users with applications such as audio-vidé player, event alarm, network storage, etc.

Service management layer provides services related to discovery. If a requested service has not been found out in local cache, a service request message is sent to forwarding manager to start service discovery.

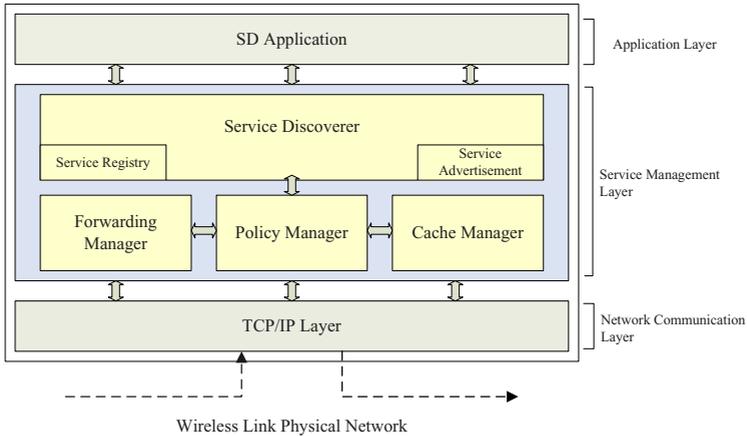


Fig. 1. Service Discovery Architecture

Forwarding manager plays a role in carrying service advertisement messages and request messages. Service discoverer picks out the shortest shortcut of neighboring nodes, by using service discovery protocol proposed in this paper to carry a request message. That is, a node with the shortest shortcut to a destination on those paths is chosen to carry service request message. Policy manager plays a role in controlling nodes through their current policies which describe service advertisement preference or replacement strategy or refresh rate, TTL(time-to-live).

Although whatever routing protocol may be used in the network communication layer, this paper uses on-demand AODV [15].

3.2 System Model

We consider a MANET that is a self-organized network of mobile nodes comprising such small mobile devices as PDAs, handheld devices, and notebook computers, which are connected with via wireless links.

The system model is as follows:

Mobile nodes are free to move at random, comprising a dynamic topology network. Each mobile node in the network has its own unique node ID. A file also has a unique file ID.

4 Proposed Service Discovery Protocol

This section describes cooperative neighbor caching of service information, the lower ID-based service discovery protocol, and cache management, for mobile ad-hoc networks.

4.1 Cooperative Neighbor Caching of Service Information

Our model is based on the concepts of peer-to-peer caching of service advertisement. A service provider saves information about services or files it provides at its local cache, then, advertises such service or file information to its neighboring node in accordance with the delivery scheme. All mobile nodes in a network serve as intermediate nodes, in accordance with the delivery scheme. This is relay messages, which are transmitted via cooperation with neighboring nodes, within the transmission range of a wireless network. Nodes in a network save and manage information received to at their local cache, for a specified period. We call this cache Service Routing Cache (SRC), in which each entry contains information such as; node Id, IP address, service name, power and ttl. Each node as a service provider acts both a server and a client when it requests a needed service. In MANET, we use AODV, which is well-known as a routing protocol in a mobile ad-hoc network. 'Hello' messages of AODV are periodical one-hop broadcast to neighboring nodes. To gather information about the neighbors of each node, we piggyback on hello messages including neighbor information, such as node ID, service name, degree of node connectivity, power and time-stamp. Therefore, we do not use separate messages to obtain information about neighbors. A neighboring node management table is generated to manage information about neighboring nodes, based on 'hello' messages obtained from them. If there is not any 'hello' message from neighboring nodes after a specified period, they are now not neighbors, because they have moved. Information about the neighboring nodes is updated in the neighboring node management table. Nodes recently joining a network inform of their existence by making a broadcast to their neighboring nodes via 'hello' messages. If there is not any 'hello' message from neighboring nodes after a specified period, they are beyond transmission range or have left the network.

4.2 The Lower ID-Based Service Discovery

The basic idea of lower ID scheme is that each node only transmits messages to neighboring nodes with lower ID, instead of broadcasting them to all neighboring node. It can be used when the node advertises its service to another node, when a service requester sends a request message to search for a needed service, and when an intermediate node transmits the received message to neighbors. A service provider saves its services in a local cache SRC, and periodically advertises them to neighboring nodes. Figure 1 shows a simple example of a lower ID scheme.

In Figure 2(a), each node in a network advertises its service to another node on the network. We suppose that Nodes 1 to 4 have Files A to D, respectively, Node 3 transmits a service advertisement message to its neighboring Node 2 of lower ID. Then, Node 2 saves information about File C and the address of Node 3 in its local cache SRC. Node 4 transmits a service advertisement message to its neighboring Node 2 of lower ID, when Node 2 saves information about File D and the address of Node 4 in its local cache SRC. Similarly, Node 2 only transmits a service advertisement message to its neighboring Node 1 of lower ID. Iteration of this advertisement process results in Node 1 having the lowest ID, which no longer transmits SADM

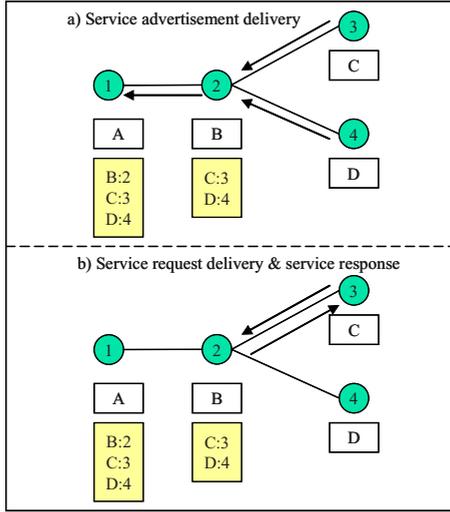


Fig. 2. Lower ID-based Service Advertisement and Service Discovery

```

SADM (msg)
//SADM: Service Advertisement Message
save service name and nodeid in local cache SRC;
for each neighbor n in neighboring node management table
{
    if n.nodeID < current.nodeID
        forward SADM to n.nodeID
}
    
```

Fig. 3. When receive service advertisement message

messages. At that time, Node 1 functions as a distributed index server for file information, producing significantly fewer messages than in case of broadcasting. This policy is associated with a service request message to search for file information. For example, Node 3 in Figure 2(b) transmits a service request message to its neighboring Node 2 of lower ID to discover File D. Node 2 contains information about File D, which implies that the search for File D is a success. In the lower ID delivery scheme, a node of lower ID acts as a server using message information transmitted in collaboration with its neighboring nodes in a distributed environment, without any centralized directory structure. This scheme is efficient because it can transmit messages to a node of lower ID without flooding in a distributed environment, reducing the transmissions and messages used for service discovery. An algorithm for services advertisement is shown in Figure 3 and we show the searching process in Figure 4.

```

SRQM (request node ID, servicename)
// SRQM: Service Request Message
// SREM: Service Response Message
//Case 1: The current node has service name
If   node has servicename in local cache SRC
    return servicename
    reply SREM;
//Case 2: A neighbor node has service name
If   one of the neighboring nodes has a service name
    return servicenam
    reply SREM;
//Case 3: A SRQM is forwarded to a node of the lower ID
for each neighbor n in neighboring node management table
{
  if  n.nodeID < current.nodeID
    forward SRQM to n.nodeID
}

```

Fig. 4. An algorithm for service discovery

While a service request message is being forwarded, or when a service requested by a service provider is found, an available node produces a service response message based on information about that service. The service response message contains the IP address of a service that can provide services. Note that there may be several service response messages. Of service providers receiving these service response messages, the one with the highest value of service power (slack time) is selected to receive a service call.

4.3 Cache Management

There is a cache consistency issue in both the lower ID scheme and the higher degree scheme. Several research works prove that a strong cache consistency is maintained in the single-hop-based wireless environment [16]. However, it is too expensive to maintain strong cache consistency in an ad-hoc network, due to its bandwidth or power constraints; in fact, a weak consistency model is more attractive [17].

To maintain cache consistency, we use a simple weak consistency model based on a time-to-live mechanism, in which, if a service entry has not exceeded its service available time, we call it “valid service entry”, it is considered to have a proper value. On other hand, if it does, we call it “invalid service entry”, it is eliminated from the local cache SRC, because services are not effective any more. If a new copy of a node has a more recent service available time, only its service available time field is updated, if not, it is dropped. If the cache size has not enough free space that a new service can not be added, service rarely referred to are deleted or eliminated in accordance with the LRU deletion policy. An algorithm for cache management is shown in Figure 5.

```
// When cache replacement is necessary;
If (not enough free space)
  If (there is invalid service entry in the cache SRC)
    then remove a invalid service entry
  else
    remove a valid service entry that has the least
    available time
```

Fig. 5. When cache replacement is necessary

5 Results and Analysis of Simulation

In this section, we evaluate the capacity of the lower ID (LD) scheme in MANET. There will be a comparison of our protocol with a flooding-based system, which is a conventional search system.

5.1 Environment of Simulation

Simulation studies are performed using NS2 [18], a representative network simulation tool. AODV protocol is used as a routing protocol, with 1500m x 1500m in the size of network area. Simulation scenarios are created with the number of nodes randomly distributed in the scenario area. The simulation is carried out with a change in the number of nodes ranging from 20 up to 100. Random way point model is used as the mobility model. A mobile scenario for mobile nodes where pause time on average is 2 seconds with average velocity and maximum velocity varying 3 to 20 m/sec and 5 to 30 m/sec, respectively, is designed for evaluating their performance. At the beginning of each simulation, some nodes are randomly selected out to act as a server. These selected servers provide randomly selected services. Table 1 shows the values of factors used in a simulation.

Table 1. Simulation Parameters

Parameter	Value
Number of nodes	20 to 100
Network Area (x, y)	1500m x 1500m
Mean query generate time (secs)	5
Hello Message Interval (secs)	5
Routing Protocol	AODV
Movement average speed (m/s)	3 to 20
Movement maximum speed (m/s)	5 to 30
Mobility Pattern	Random way-point
Duration	300s
Transmission range	250m
Server Rate	30%
Mac	Mac/802_11

5.2 Results of Simulation

The average search hop count, a measurement standard that evaluates algorithm performance is the number of average hops needed for the search. More nodes cache more related information, locality and service response will get higher and quicker, respectively. When a service requester found out available services, the number of physical hops for paths is reduced, it can quickly find them out without going through many nodes and that power of node is relatively less consumed. Figure 6 shows physical hop count from source node to destination node.

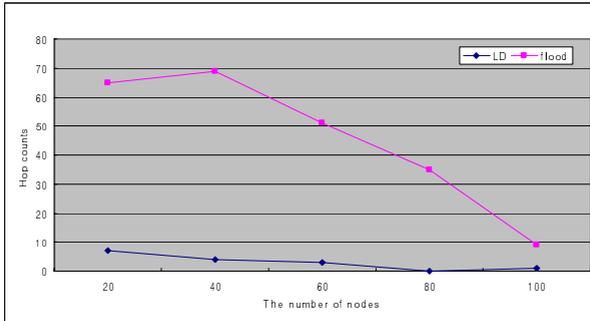


Fig. 6. Average physical hop count from source node to destination node

Figure 7 shows the number of messages nodes communicate. In flooding method, a number of messages are communicated by broadcasting messages for service advertisements and service searches to neighboring nodes, which means that a transmission of messages is delayed with nodes consuming lots of power.

To inspect the effects of node speed, we run simulation sets that use the two selected service discovery protocols, respectively. In these simulations, the number of nodes is fixed to 100. Each set includes four subsets of simulations, where average velocity is set to 5 m/s, 10 m/s, 15 m/s, 20 m/s, respectively. Figure 8 shows the effect

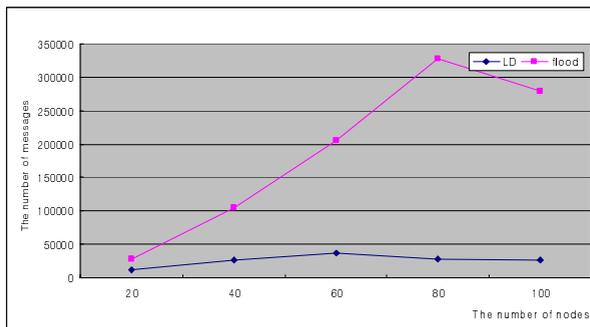


Fig. 7. Message count of sending and receiving

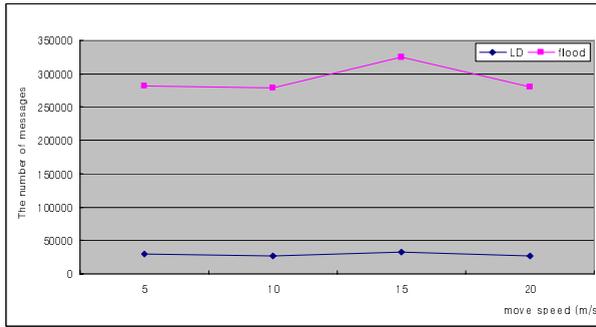


Fig. 8. Message count of sending and receiving under different node speed

of node speed on the number of messages of sending and receiving. LD scheme has the lower message under different node speed. Thus, LD scheme is effective protocol under different node speed.

In our proposed method, the number of messages communicated among nodes is a little more than flooding-based method, while the proposed method does not incur countless messages unlike flooding-based service discovery.

6 Conclusion

In this paper, we propose a service discovery scheme to effectively discover services by using only local information. The proposed scheme is suitable for a mobile ad-hoc network, where mobile devices with such features as high mobility and resource constraints comprise wireless networks without any infrastructure. While the flooding scheme incurs high costs in MANET, since there are lots of messages exchanged, our proposed scheme shows in simulation results that it is very effective in significantly reducing physical hop counts and the number of messages exchanged. A reduction in physical hop counts means reducing query delay by reducing response time, and a reduction in the number of messages exchanged implies a reduction in total network traffic costs and node workloads. Since nodes in MANET dynamically move and the neighboring nodes are changed in their transmission range. The simulation results show that, in the proposed scheme, physical hop counts and the number of messages exchanged have been significantly reduced, compared with the other protocol.

References

1. Toh, C.K.: Ad Hoc Mobile Wireless Networks. Protocols and Systems. Prentice-Hall, Englewood Cliffs (2002)
2. Meier, R., Cahill, V., Nedos, A., Clarke, S.: Proximity-Based Service Discovery in Mobile Ad Hoc Networks. In: Kutvonen, L., Alonistioti, N. (eds.) DAIS 2005. LNCS, vol. 3543, pp. 115–129. Springer, Heidelberg (2005)
3. The Gnutella web site, <http://www.gnutella.com>
4. The KaZaA web site, <http://www.kazaa.com>

5. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 conference on Applications technologies architectures and protocols for computer communications, pp. 149–160. ACM Press, New York (2001)
6. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content addressable network. In: Proceedings of the 2001 conference on Applications technologies architectures and protocols for computer communications, pp. 161–172. ACM Press, New York (2001)
7. Rowstron, A.I.T., Druschel, P.: Pastry Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
8. Jini Architectural Overview: Technical White Paper, <http://www.sun.com/software/jini/whitepapers/architecture.html>
9. Guttman, E.: Service Location Protocol Automatic Discovery of IP Network Services. IEEE Internet Computing 3, 71–80 (1999)
10. White paper, Salutation Architecture overview (1998), <http://www.salutation.org/whitepaper/originalwp.pdf>
11. Helal, S., Desai, N., Verma, V., Lee, C.: Konark-A Service Discovery and Delivery Protocol for Ad-hoc Networks. In: Proc. of the Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans (March 2003)
12. Helal, O., Chakraborty, D., Tolia, S., Kushraj, D., Kunjithapatham, A., Gupta, G., Joshi, A., Finin, T.: Allia Alliance-based Service Discovery for Ad-Hoc Environments. In: Second ACM International Workshop on Mobile Commerce, in conjunction with Mobicom 2002, Atlanta GA, USA (2002)
13. Understanding UPnP. A White Paper, Microsoft Corporation (June 2000)
14. Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: Toward Distributed Service Discovery in Pervasive Computing Environments. IEEE Trans. on Mobile Computing 5(2), 97–112 (2006)
15. Perkins, C.E., Royer, E.M., Das, S.: Ad Hoc On-Demand Distance Vector Routing (AODV) Routing. RFC 3561, IETF (July 2003)
16. Cao, G.: Proactive Power-Aware Cache Management for Mobile Computing Systems. IEEE Trans. Computer 51(6), 608–621 (2002)
17. Yin, L., Cao, G.: Supporting Cooperative Caching in Ad Hoc Networks. IEEE Trans. on Mobile Computing 5(1), 77–89 (2006)
18. NS2 Object Hierarchy, <http://www-sop.inirria.fr/planete/software/ns-doc/ns-current/aindex.html>