

A cluster-based web service discovery in MANET environments

Yeon-Seok Kim, Yoo-Seok Shim and Kyong-Ho Lee*

Department of Computer Science, Yonsei University, 134, Shinchon-dong, Sudaemoon-ku, Seoul 120-749, South Korea

Abstract. As standard technologies to implement a service-oriented architecture, Web services support interoperability between heterogeneous platforms. In the ubiquitous era, for Web services to become a universal software development paradigm, they must be able to support a MANET environment with a variety of mobile devices. In this paper, we propose an efficient method that discovers services in MANET environments, where mobile devices are free to move independently. The proposed method constructs stable clusters based on the mobility of devices. It also selects an appropriate service discovery scheme for a cluster depending on its characteristics. Experimental results show that the proposed method outperforms previous methods, and discovers services based on the features of a cluster.

Keywords: Web service discovery, clustering, mobility, mobile ad-hoc network

1. Introduction

Web services have emerged as standard technologies, which can implement a service-oriented architecture (SOA) [4,25]. Specifically, Web services standards are designed to support interoperability between heterogeneous platforms. However, for Web services to become a universal paradigm for systems integration, they must be able to support a MANET (Mobile Ad-hoc NETwork) [2,17,26] environment where the network topology is dynamically changing by moving devices.

MANET consists of a set of wireless mobile devices or nodes (hereafter nodes), which communicate with each other without centralized control or established infrastructure. In particular, each node in MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. For effectively supporting Web services in this environment, it is necessary to discover services efficiently [22,23]. The proposed method constructs stable clusters by considering the mobility of nodes, and selects an appropriate service discovery scheme for a cluster depending on its characteristics. Since clustering is generally effective for the scalability of a mobile network, it is very useful for a MANET environment [1].

The proposed method focuses on the mobility of nodes in clustering them to support the efficient discovery of their services in a MANET environment. This is because nodes usually move in a certain direction in the real world. In other words, mobile nodes, which include smart phones carried by people or onboard units on cars, often move according to a certain constraint of an environment. For example, if people move in a particular area or drive a car, their mobility is determined by the rules set by the

*Corresponding author. Tel.: +82 2 2123 3878; Fax: +82 2 365 2579; E-mail: khlee@cs.yonsei.ac.kr.

society. Therefore, if we consider the mobility of nodes including moving direction [12], it is possible to construct a more stable cluster.

Several methods for discovering services in MANET have been proposed. Charkaborty et al. [9] propose a service discovery protocol, which is based on the concepts of peer-to-peer caching of service advertisements for a dynamic environment. Lua et al. [19] propose a distributed hash table (DHT) architecture for a static environment. However, since MANET can be variable depending on the feature of constituent nodes, it is not suitable to apply one and only one discovery architecture. In this paper, we construct a stable cluster by analyzing the mobility of nodes, and select an appropriate service discovery architecture depending on the feature of a cluster. The proposed mobility-based clustering and dynamic service discovery include the following features.

- Constructing stable clusters by considering the mobility of nodes
- Reducing the frequent participation and departure of nodes by determining the criteria for participation in a cluster
- Overcoming the heterogeneity of mobile devices by employing Web services technologies
- Selecting an appropriate service discovery scheme depending on the feature of a cluster

To evaluate the performance of the proposed method, we compared the proposed method with previous ones in terms of the number of the clusters created and deleted, and the number of the messages exchanged. Specifically, we evaluated whether the proposed method selects an appropriate service discovery scheme depending on the feature of clusters. Experimental results show that the proposed method outperforms previous methods.

The organization of this paper is as follows. First, a brief survey on related work is presented. Next, the proposed method for clustering nodes and discovering services is described in detail. Next, the performance of the proposed method is analyzed and compared with the conventional method. Finally, conclusions and future works are summarized.

2. Related work

Various clustering methods have been proposed for scalability in MANET. However, some of them do not achieve good performance when mobile nodes move frequently. This is because they only consider a limited number of factors such as the connectivity between nodes, the capability of a cluster, and nodes' link duration [6,13,18,28,30]. This section presents a brief survey of conventional methods as summarized in Table 1.

Yang and Chang [6] propose a stability-based clustering method. The stability of a node is measured by the number of its neighbor nodes. A cluster head, however, can be changed frequently due to the change of the stability. Srirama et al. [24] propose a mobile Web service discovery approach, which uses JXTA modules for publishing and discovering services in P2P networks. The method classifies constituent nodes into four categories: super peer, edge peer, rendezvous peer, and relay peer.

Marin-Perianu et al. [20] propose an energy-efficient service discovery protocol and a lightweight clustering algorithm for service registration in wireless sensor networks. The clustering algorithm is based on one-hop neighborhood information and if a service does not exist in a cluster, the method discovers services by sending a request message to an adjacent cluster head. However, since the method only considers the capabilities of nodes for selecting a cluster head, cluster heads can be changed frequently with the high mobility of nodes in MANET. Meier et al. [21] propose a mobile agent-based service discovery technique. The method defines two agents: travel and stationary agents. Travel agents

Table 1
Clustering methods for discovering services in a mobile environment

Author	Year	Feature
Yang and Chang [6]	2008	Cluster nodes based on their stability
Srirama et al. [22]	2008	Propose a mobile Web service discovery mechanism based on the features of P2P networks
Marin-Perianu et al. [20]	2008	Propose an energy-efficient service discovery protocol for heterogeneous wireless sensor networks
Meier et al. [21]	2008	Develop a mobile agent-based scheme for service discovery
Gao [28]	2007	Presents a multi-hop cluster-based architecture for service discovery
Tolba et al. [13]	2007	Cluster nodes based on their mobility, connectivity, and energy
Xie et al. [18]	2007	Propose a velocity-aware routing protocol
Artail et al. [14]	2007	Use a cluster head as a service directory node for service discovery
Obaid et al. [3]	2007	Describe a service discovery strategy coupled with routing protocol
Liang et al. [16]	2007	Discover services by applying DHT to cluster heads
Bazzal et al. [30]	2006	Construct clusters based on the capacity and link lifetime of nodes
Gao et al. [31]	2006	Introduce an enhanced protocol to reduce the packet overhead of GSD
Yang et al. [29]	2006	Presents a service discovery protocol based on multicast query and reply
Charkaborty et al. [9]	2006	Propose a distributed service discovery protocol based on the concepts of P2P caching and group-based forwarding

(TAs) are roaming throughout the network and collecting dynamically changing service information. Meanwhile, stationary agents (SAs) on each node are managing a list of the available services and adjust the number of TAs. Specifically, the appropriate number of TAs depends on the dynamics in the network, i.e. how frequently nodes appear or disappear in MANET. To take this factor into account, SAs determine the so-called Churn Rate (CR), which describes to which degree neighboring nodes have changed in a certain time interval.

Gao [28] proposes a multi-hop cluster-based architecture for service discovery (CASD) in a ubiquitous computing network. CASD organizes the network to be multi-hop clusters with better stability and flexibility. It selects cluster heads based on nodes' neighborhood benchmark scores, which quantify the connectivity and link stability of mobile nodes. Tolba et al. [13] propose a connectivity, energy and mobility driven clustering algorithm (CEMCA), which considers the node connectivity, battery energy and mobility of nodes. Each node has a quality that indicates its suitability as a cluster head. CEMCA selects nodes with the high suitability as cluster heads.

Xie et al. [18] propose a threshold-based hybrid routing protocol, which considers switching between the Proactive MANET Protocol (PMP) and the Reactive MANET Protocol (RMP) according to the current velocity of a node and the network situation. The method manages slow and fast moving nodes by using PMP and RMP, respectively. Specifically, the threshold for the protocol switch is based on the difference of velocity, the duration of the current velocity, and the number of existing nodes within a proactive cluster.

Artail et al. [14] propose a decentralized and scalable service discovery framework for ad-hoc networks. The framework uses cluster heads as service directory nodes for finding desired services. Obaid et al. [3] propose a protocol for service discovery on top of a reactive routing protocol. Specifically, it discovers services by using the service advertisements and caches of each node. However, it can increase the cost of a network because it does not consider a cluster.

Liang et al. [16] propose a Mobile Service Discovery Protocol (MSDP), which uses a dynamic clustering algorithm to group nodes in MANET, and utilizes a distributed hash table (DHT) to efficiently cache service information in a peer-to-peer manner. MSDP constructs clusters by using a lightweight clustering algorithm and discovers services by applying DHT to cluster heads. However, for the selection of a cluster head, MSDP uses the lowest ID clustering approach, which does not consider the features

of a node. Bazzal et al. [30] propose a Flexible Weight based Clustering Algorithm (FWCA), which minimizes the number of invocations of the algorithm and maximizes the lifetime of a cluster. FWCA utilizes such factors as the node's capacity and the link lifetime between nodes for constructing a cluster.

Gao et al. [31] propose an enhanced version of GSD [8] called PCPGSD, which reduces the packet overhead of GSD. PCPGSD consists of three mechanisms: Pruning of Far Candidate Nodes (PFCN), Combining of Relay Nodes (CRN), and Piggybacking of Relay Nodes (PRN). To reduce the number of request packets, PFCN omits all candidate nodes that are too far to be reached by the request packet, and CRN gets a relay node to process service requests. In the PRN case, the list of relay nodes is piggybacked in the request packet. Thus, only one request packet sent in broadcast mode is enough. However, this method is inefficient in an environment with high mobility.

Yang et al. [29] present a High Efficiency Service Discovery protocol (HESED) for efficient Web service discovery. HESED is based on multicast query and reply. Specifically, all nodes cache received service information. However, the protocol can increase a network overhead for service discovery because the method does not cluster nodes and supports a routing by edge node forwarding. Charkaborty et al. [9] propose a distributed service discovery protocol based on the concepts of peer-to-peer caching of service advertisements and group-based intelligent forwarding of service requests. Services are described using the Web Ontology Language (OWL) [11] and the service descriptions are cached by advertisements.

As was seen above, most previous works on service discovery in MANET do not sufficiently consider the dynamic nature of the network. They also use limited features such as the connectivity between nodes, the capability of each cluster, and nodes' link duration. In this paper, we propose an efficient cluster head selection algorithm for maximizing the duration of a cluster in a MANET environment. The proposed method selects a cluster head by sophisticatedly analyzing the mobility of nodes. Therefore, the duration of a cluster head is maximized, and the network is maintained and managed at low cost. Moreover, while previous works employ a uniform service discovery, the proposed method dynamically selects a service discovery scheme depending on the features of clusters.

3. The mobility-based clustering

In this section, we describe the proposed mobility-based clustering method in detail. Generally, clustering is represented by the selection process of a cluster head. The proposed method defines the mobility of nodes as the basis of constructing stable clusters. The method selects a cluster head as shown in Fig. 1. First, each node performs a mobility evaluation, where collects the mobility information of neighboring nodes within two hops and checks whether the similarity among their mobility exists. If there is no similarity among the mobility of nodes, connectivity values with neighboring nodes are evaluated. The suitability of the node as a cluster head is computed by using the results of the mobility and connectivity evaluations. The proposed method selects a node with the highest suitability as a cluster head.

3.1. Analyzing the mobility of nodes

To analyze the mobility of nodes, the proposed method considers the moving direction and velocity of nodes, their sustained duration, and the connectivity with neighboring nodes. The factors to define the mobility of nodes and their effects on clustering are as follows.

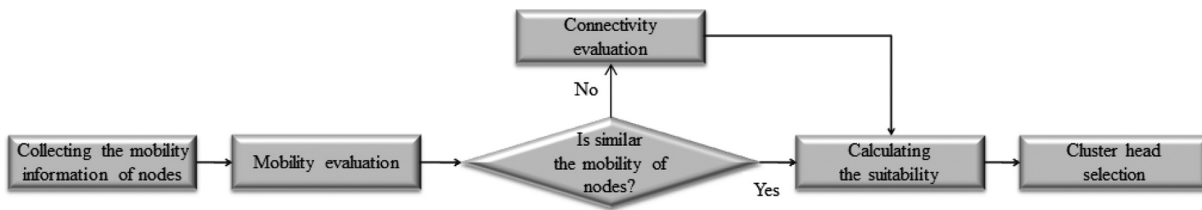


Fig. 1. The proposed method of selecting a cluster head.

- Moving direction and velocity: If nodes in a network have similar direction and velocity one another, they may form a more stable cluster.
- Sustained duration: If a node in a network maintains the current movement for a sustained duration, it may form a more stable cluster head.
- Connectivity: If a node in a network has more adjacent nodes as its neighbors, it may form a more stable cluster head.

Based on the above considerations, the proposed method analyzes the movement of nodes that change over time. Based on the mobility analysis, the proposed method selects cluster heads and discovers services in MANET. In this paper, we assume that every node is able to compute the values of the above mobility items. The mobility of node n is defined by

$$\text{Mobility}(n) = (\text{direction}, \text{velocity}, \text{duration}, \text{connection}) \quad (1)$$

Direction and velocity are approximate values to the movement properties of an actual node and are determined within tolerances. Duration is the sustained period of time of a specific direction and velocity, and connectivity denotes the number of adjacent neighboring nodes. If a node moves without any significant change, its mobility remains valid. Otherwise, the mobility value of a node changes accordingly. If the factors do not represent the mobility of a node due to its irregular movement, the mobility value is not set. There may be nodes in a network, which do not have the mobility value.

3.2. Selecting a cluster head

A cluster head plays a key role in constructing and maintaining a cluster and discovering services. It is important that a stable node, which can play the role of a cluster head for the longest time, should be selected as a cluster head. The proposed method selects cluster heads appropriately depending on the mobility of nodes in a network.

The proposed method computes the suitability as a cluster head (hereafter, the suitability) for each node by using the mobility and connectivity evaluation. The suitability means the suitability as a cluster head of a node. The node with the highest suitability value is selected as a cluster head. To compute the suitability, the proposed method uses both the mobility and connectivity evaluation. The procedures of the suitability computation for selecting cluster heads are as follows.

First of all, all nodes collect the mobility information of their neighboring nodes within two hops. After collecting the mobility information, each node determines the representative moving direction of neighboring nodes. Candidate nodes, whose directions are similar to the representative moving direction, are considered to perform the mobility evaluation. Specifically, the proposed method selects two adjacent quadrants with most directional distributions in two-dimensional plane, on which the movement direction of neighbor nodes are represented. After that, the method computes the average direction of the nodes

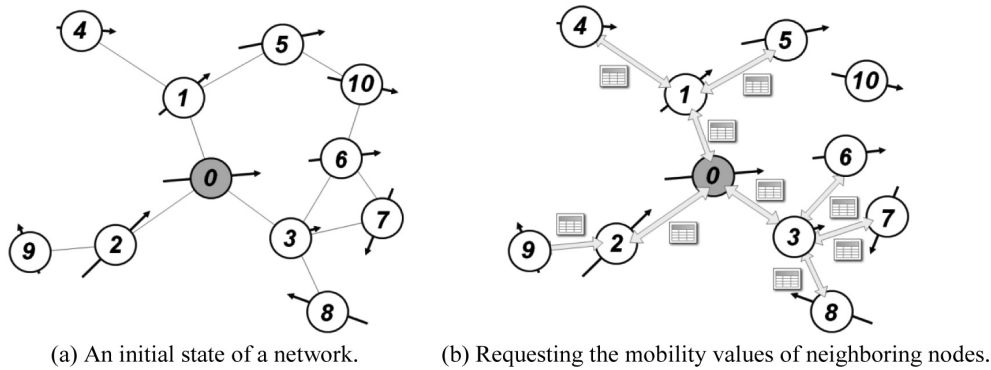


Fig. 2. An example of a MANET environment for constructing a cluster.

No	Mobility
1	(40°, 29, 6, 3)
2	(47°, 42, 5, 2)
3	(16°, 22, 4, 6)
4	(-10°, 21, 5, 1)
5	(8°, 39, 4, 1)
6	(2°, 32, 5, 3)
7	(248°, 34, 3, 2)
8	(149°, 40, 2, 1)
9	(104°, 22, 3, 1)

Fig. 3. An example of the mobility information collected from the neighboring nodes of node 0.

that belong to the two quadrants, that is, an initial representative direction, and sets the final representative direction through a correction step. In the correction step, the average direction of nodes within ± 90 degrees from the initial representative direction is computed and set as the final representative direction. Finally, the proposed method selects the nodes within ± 90 degrees of the final representative direction as the candidate nodes for the mobility evaluation.

The following is an example of constructing a cluster by the proposed method. In Fig. 2, a circle and a number represent a node and its unique identifier, respectively. Also, the size and direction of an arrow indicate the speed and moving direction of the node. Figure 2(a) shows an initial state of a network. In Fig. 2(b), node 0 (the shaded node) requests the mobility information of its neighboring nodes. Figure 3 illustrates the mobility information of the neighbors of node 0. In this example, we assume that the mobility values of nodes 0 and 10 are $(4^\circ, 44, 3, 3)$ and $(-24^\circ, 32, 2, 2)$, respectively.

As mentioned previously, all nodes collect the mobility information from neighboring nodes within two hops. Based on the representative moving direction of their neighbors, the proposed method selects candidate nodes, to which the mobility evaluation would be applied. Figure 4 shows the distribution of moving direction of neighboring nodes. To determine a representative direction, the proposed method computes an initial representative direction by computing the average direction from nodes 1 ~ 3, 5, 6, 8, and 9, whose directions are located at quadrants 1 and 2 with the highest direction distribution. After that, the method computes the average direction from nodes 1 ~ 6 and 9, whose directions are within ± 90 degrees from the initial representative direction. Finally, we select nodes within ± 90 degrees of the

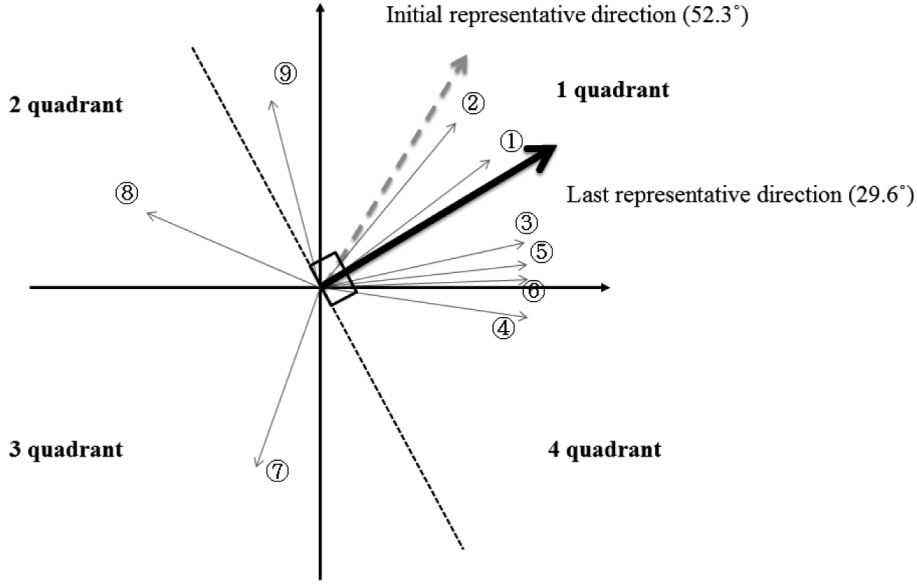


Fig. 4. Determining the final representative direction for the neighbors of node 0.

final representative direction as candidates for the mobility evaluation.

The mobility evaluation is performed to compute the mobile stability of the candidate nodes. The mobile stability of a node indicates how its movement is close to the average mobility of its neighboring nodes. The mobility evaluation considers the similarities to a representative direction and an average velocity, and a ratio to the maximum duration of a sustained movement. The mobile stability of node i , $mStability(i)$, is calculated by

$$mStability(i) = w_1 \times SimilarityToRepresentativeDirection(i) + w_2 \times SimilarityToAverageVelocity(i) + w_3 \times RatioToMaximumDuration(i) \tag{2}$$

$$where, SimilarityToRepresentativeDirection(i) = 1 - \frac{|Dir(i) - Avg_{dir}|}{90^\circ},$$

$$SimilarityToAverageVelocity(i) = 1 - \frac{|Vel(i) - Avg_{vel}|}{MaxdiffFromAvgVel}, \text{ and}$$

$$RatioToMaximumDuration(i) = \frac{Dur(i)}{MaxDur}.$$

$$(MaxDiffFromAvgVel = Max(|Vel(1) - Avg_{vel}|, \dots, |Vel(n) - Avg_{vel}|),$$

$$MaxDur = Max(Dur(1), Dur(2), \dots, Dur(n)),$$

n : the number of candidate nodes, Avg : the average value of each factor, $\sum w_i = 1$)

Figure 5 shows the result of the mobility evaluation for the neighboring nodes of node 0. The node with the most similar mobility to the average has the highest $mStability$. In the case of node 2, although its moving direction is similar to the representative direction, it has the $mStability$ of a relatively low value. This is because its velocity is far faster than the average velocity. Figure 6 shows the result of the mobility evaluation for all the neighbors of each node.

No	Mobility	(Direction, Velocity, Duration similarity)	mStability
1	(40°, 29, 6, 3)	(0.884, 0.954, 1.000)	0.946
6	(2°, 32, 5, 3)	(0.694, 0.805, 0.833)	0.777
3	(16°, 22, 4, 6)	(0.849, 0.391, 0.667)	0.636
4	(-10°, 21, 5, 1)	(0.560, 0.310, 0.833)	0.568
5	(8°, 39, 4, 1)	(0.760, 0.241, 0.667)	0.556
2	(47°, 42, 5, 2)	(0.806, 0.000, 0.833)	0.547
9	(104°, 22, 3, 1)	(0.173, 0.391, 0.500)	0.355

Fig. 5. An example of the mobility evaluation for the neighboring nodes of node 0.

No	mStability	No	mStability	No	mStability	No	mStability	No	mStability
0	0.568	0	0.363	0	0.464	0	0.517	0	0.494
2	0.617	1	0.991	1	0.627	1	0.583	1	0.791
3	0.593	3	0.632	2	0.569	5	0.788	4	0.614
4	0.604	9	0.436	6	0.746	(d) The case of node 4.	6	0.932	
5	0.776			10	0.483		10	0.669	
10	0.650	(b) The case of node 2.		(c) The case of node 3.		(e) The case of node 5.			
(a) The case of node 1.									
No	mStability	No	mStability	No	mStability	No	mStability	No	mStability
0	0.640	0	0.517	0	0.521	0	0.454	1	0.854
3	0.611	3	0.568	3	0.588	2	0.587	3	0.554
5	0.845	6	0.976	6	0.961	(i) The case of node 9.	5	0.524	
10	0.680	10	0.698				6	0.832	
(f) The case of node 6.		(g) The case of node 7.		(h) The case of node 8.		(j) The case of node 10.			

Fig. 6. An example of the mobility evaluation for all the neighbors of each node.

After the mobility evaluation, to check the similarity among the mobilities of nodes, the standard deviation of the mobile stability values of neighboring nodes is considered. If the standard deviation is less than a threshold, the proposed method considers the similarity to be high and computes the suitability of a node as a cluster head based on its mobile stability. Otherwise, the suitability is computed from the connection evaluation. Meanwhile, the threshold is set depending on the state of a network. If the threshold is set to be small, the suitability would be computed by the connectivity evaluation rather than the mobility evaluation.

Meanwhile, the connectivity evaluation is done to determine the connectivity value between nodes. The connection value of a node is defined as the number of its neighboring nodes. Namely, the connectivity of node i , $Connectivity(i)$, is defined as the ratio of its connection value to the maximum connection value of its neighbors as shown in Eq. (3). This connectivity evaluation selects the most appropriate cluster head in terms of the connectivity of a node.

$$Connectivity(i) = \frac{NumOfNeighbors(i)}{\max_{j=1}^n (NumOfNeighbors(j))} \quad (3)$$

(n : the number of neighbor nodes within two hops)

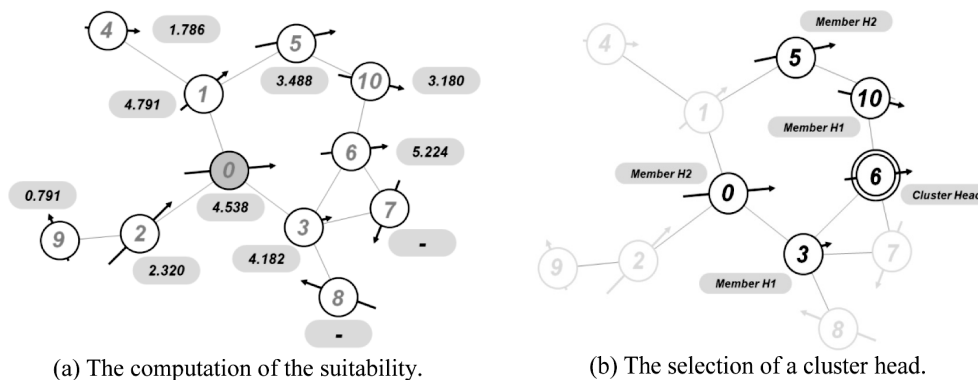


Fig. 7. A clustering result of the network of Fig. 2.

After computing the mobile stability or connectivity of all nodes, the suitability of each node is computed. Then, the method selects the node with the highest suitability as a cluster head. This process is executed repeatedly when the configuration of a cluster is destroyed or changed due to the mobility of nodes.

Figure 7 shows an example of selecting a cluster head. Figure 7(a) represents the suitability of each node, which is determined by adding up the mobile stability values computed by its neighboring nodes. For example, the suitability of node 0 comes to 4.538 ($= 0.568 + 0.363 + 0.464 + 0.517 + 0.494 + 0.640 + 0.517 + 0.521 + 0.454$) by adding up the mobile stability values of node 0 computed by its neighboring nodes 1 ~ 9. However, nodes 7 and 8 do not have the suitability. This is because they are excluded from the mobility evaluation due to the difference from the representative direction of their neighboring nodes. Finally, the proposed method determines node 6 with the highest suitability as a cluster head like Fig. 7(b). Nodes 7 and 8 are also excluded from the cluster configuration.

4. Service discovery

This section introduces how to discover services based on the clusters constructed. The proposed method selects an appropriate discovery architecture according to the feature of a cluster for an efficient service discovery.

4.1. Selecting a discovery architecture

Service discovery architectures may have advantages and limitations in MANET with a dynamic topology. For example, flooding-based discovery processes all discovery requests by an on-demand scheme. While this approach does not require any structure, it needs a high cost to discover services. On the other hand, while a DHT (decentralized hash table)-based approach discovers services quickly, it needs a great cost to configure an initial discovery environment because it distributes and manages the key values of services throughout a network. To reflect the different natures of discovery architectures, we select an appropriate discovery architecture depending on the feature of a cluster.

The proposed method selects a discovery architecture from the three conventional architectures; flooding, centralized hash table (CHT) [27], and DHT. The brief description of each discovery architecture is as follows. In a flooding-based approach, the service request of a node is sent to its adjacent neighboring nodes. This approach has a good performance in an environment with the small number of nodes. Among

numerous flooding strategies, we adopt a blind flooding approach [5], in which a request node broadcasts a query message to its neighbors without considering the efficiency of message delivery. Blind flooding is useful when a cluster is created or when nodes join or depart from a cluster.

DHT is used to store an index distributed among nodes in a self-organizing overlay network. While DHT-based approaches cost a great deal to configure a discovery environment, it takes less cost for the service discovery. The proposed method uses Chord [15], where service keys are assigned an m -bit identifier using what is known as consistent hashing. The base hashing function for consistent hashing is the SHA-1 algorithm. Consistent hashing is efficient as nodes join and leave a network without disrupting the network.

CHT is a discovery architecture, which keeps a list of services in a cluster head. While the construction of CHT is more complex than flooding, the discovery through a cluster head needs a low overhead. The proposed method represents a service as a pair of a key and a value in the hash table of a cluster head. The key is a 128-bit integer and is computed by applying the SHA-1 hash function to a service description. Also, the value contains how to access a service node. When a cluster is initially constructed or new nodes join a cluster, each node have to register its services and their addresses in the hash table of a cluster head since CHT searches for a service by using the hash table in a cluster head. Since CHT may cause a bottleneck, it is not suitable if a cluster consists of a large number of nodes and involves the frequent request of services.

The proposed method to select a discovery architecture is as follows. At first, a cluster head chooses flooding as an initial discovery architecture. This is because flooding needs no specific process to publish services. Furthermore, the proposed method needs time to calculate the frequency of service requests. Therefore, initially, a cluster discovers services using flooding for a certain period of time and calculates the number of the service requests. After computing the frequency of service requests, the proposed method estimates the number of messages that each discovery architecture may generate by considering the frequency of service requests, the number of nodes in a cluster, and the number of nodes that join or leave a cluster. Finally, the method determines a discovery architecture, which requires the fewest messages, as the new discovery architecture of a cluster. Equations (4) and (5) represent the numbers of messages required by flooding and DHT, respectively. On the other hand, if there is no sufficient difference between flooding and DHT, this corresponds to the cases where the number of nodes is relatively small or nodes join and leave frequently. Particularly, in the first case where a cluster contains the small number of nodes, the proposed method selects CHT, through which we can resolve the network bottleneck.

$$MessageInFlooding = Num_{ServiceRequests} \times n^2 \quad (4)$$

(n : the number of nodes in a cluster, $Num_{ServiceRequests}$: the frequency of service request)

$$MessageInDHT = NumberOfChangingNodes \times \log^2 n + Num_{ServiceRequests} \times \log n \quad (5)$$

($NumberOfChangingNodes$: the number of nodes that join or leave a cluster)

4.2. Publishing and discovering services

After a cluster head selects a service discovery architecture, nodes perform an appropriate service publication process. In the case of flooding, a node does not publish any services since services are discovered by searching all nodes directly. In the case of CHT, however, a node sends the list of its

own services to its cluster head for the service publication. In the case of DHT, services are published by using the processes defined by the Chord algorithm. Namely, a node computes a key value from its service description by using the SHA-1 hash function and also determines a node, which contains the reference to the service, by using a routing table called a finger table.

The service discovery process is more complex than publication. First, a node, which wants to find a specific service, has to determine whether it belongs to a cluster or finds a cluster head among its neighbor nodes. If it does not belong to any cluster and there is no cluster head in its neighborhood, it is impossible to discover services. So it has to wait until it detects a cluster head. If it belongs to a cluster or finds cluster heads, it can figure out an architecture type with the help of its cluster head node.

In the case of flooding, a node sends request messages to adjacent nodes directly. Upon receiving request messages, each node checks the existence of the service requested and then passes along to next nodes in the same cluster, which can be done simply by confirming that they belong to the same cluster head. In the case of CHT, a node can discover services through its cluster head by querying the request message. Finally, in the case of DHT, after a request node obtains a key value for a service description like CHT, then computes the location of a node with the service by using a pre-defined key space.

5. Experimental results

To evaluate the performance of the proposed method, we have experimented on two different modes of mobility. Specifically, to simulate a situation where nodes move randomly, we employed the Random Way Point (RWP) [7], which is a well-known and widely used mobility model in MANET. On the other hand, in order to simulate a network, in which nodes follow certain moving patterns, we expended the conventional Location Model [10] to a proposed model called the Extended Location Model (ELM).

The Location Model extends RWP in that when a node chooses its initial location as well as any new location it wishes to travel to, it no longer picks the location at random but instead chooses a location from a discrete set of predetermined locations or points of interest. Unlike the Location Model, ELM enables us to specify areas of certain sizes rather than points. The adjustment of the size of an area makes it possible to control the similarity level of the node movement. In order to consider a network, in which nodes have certain directivity, we could control the size of interesting areas. If an area is set to be small, the moving patterns of the nodes moving to the area become similar. Figure 8 shows the two mobility models used in the proposed method. Figures 8(a) and 8(b) represent RWP, where nodes are free to move, and ELM, where nodes move with a certain moving pattern between pre-defined areas, respectively.

The experimental environment is as follows. Initially, nodes exist in any location and move freely according to pre-defined mobility models. Each node has a certain number of services, and discovers services continuously. Table 2 shows the experimental environment. The value of the threshold used for the mobility analysis was experimentally determined. As a result, the creation and deletion of clusters were minimized when the threshold was set to 0.8 in RWP and 0.5 in ELM, respectively. To evaluate the performance of the proposed method, we measured the frequency of the creation and deletion of clusters as well as the number of the messages transmitted.

5.1. Performance analysis: Clustering

To evaluate the performance of the proposed clustering method, we measured the numbers of clusters, which were created and deleted, and the messages transmitted. Particularly, the performance of the

Table 2
Experimental environment

Criteria	Value	Criteria	Value
the number of nodes	200/500	the velocity of nodes (ELM)	0.5 ~ 2.0
network size	1,000 × 1,000	the number of interesting areas (ELM)	8
transmission distance	100	the radius of interesting areas (ELM)	50
mobility model	RWP, ELM	the number of services in a node	5 ~ 10
the moving direction of nodes (RWP, ELM)	0° ~ ± 180° or 0° ~ 360°	the frequency of service requests	the service discovery by 10% chance per unit time
the velocity of nodes (RWP)	0.5~ 2.0		

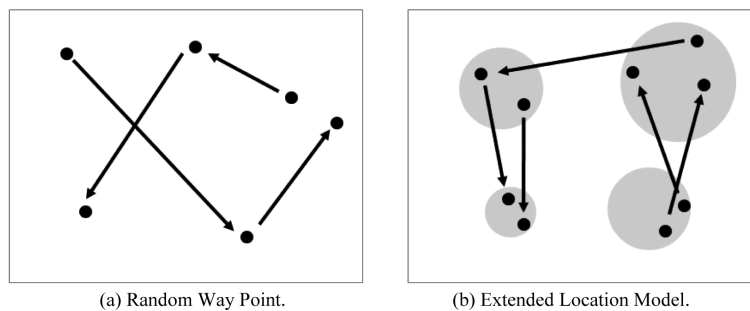


Fig. 8. RWP and ELM.

proposed method is compared with the Connectivity, Energy and Mobility driven Clustering Algorithm (CEMCA) [13]. CEMCA selects a cluster head in an attempt to minimize a routing overhead in MANET. Since it constructs clusters based on the mobility and connectivity of nodes, it is one of the most comparable works with ours. CEMCA defines the mobility of a node as the changed distance from neighboring nodes for a certain period of time. It also selects a cluster head based on the number of neighboring nodes and the energy level of a battery. Figure 9 shows the experimental result in terms of the number of the clusters created and deleted.

The experimental result showed that the proposed method created and deleted more clusters than CEMCA in the case of RWP. This is because the proposed mobility evaluation was not performed due to the irregular movements of nodes. On the other hand, in the case of ELM, the proposed method created and deleted fewer clusters than CEMCA. This indicates that the proposed method is more efficient when nodes have a specific movement pattern. From the experimental result, we can say that the proposed method is more efficient in the case where the movement of mobile devices and nodes has a certain level of similarity.

Figure 10 shows the total number of the messages required to maintain clusters during 400 seconds in the case of ELM. The proposed method produced more messages than previous clustering algorithms in an early stage. This is because, as mentioned before, the proposed method collects the mobility information of adjacent nodes initially. However, by constructing stable clusters through the mobility analysis, the amount of messages was reduced gradually. While the proposed method requires more messages in an early stage due to the mobility analysis, the stability of clusters reduces the number of messages gradually, which are needed to maintain clusters after the stable cluster creation.

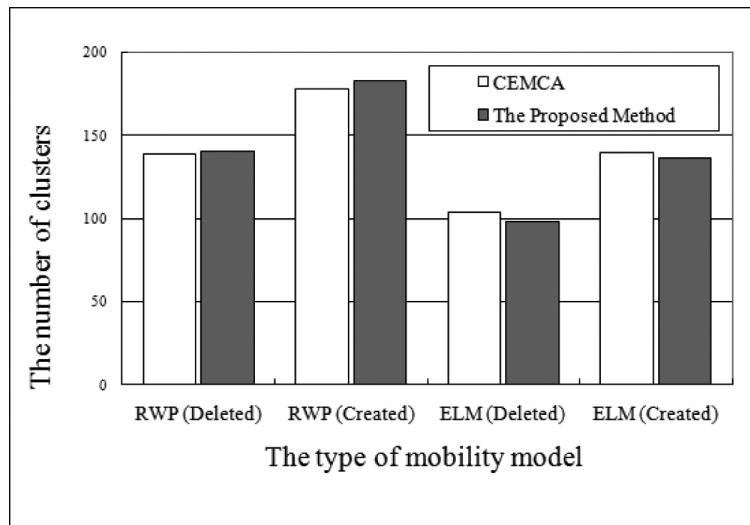


Fig. 9. The number of the clusters created and deleted.

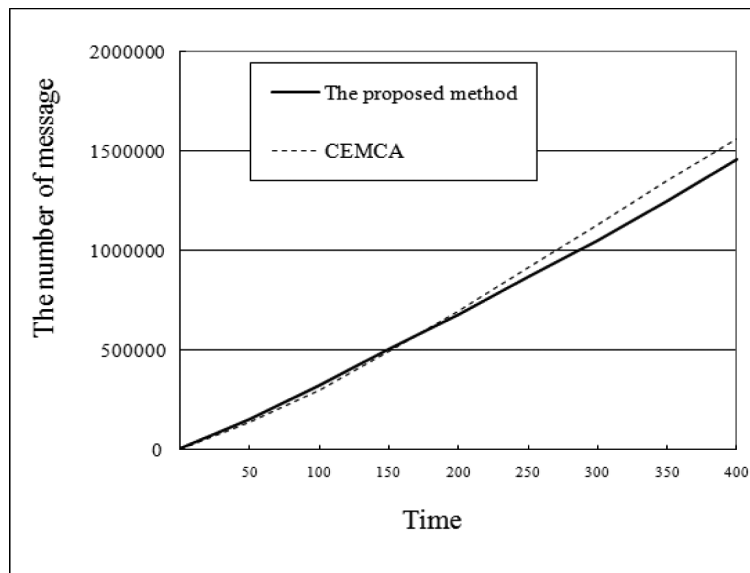


Fig. 10. The number of message.

5.2. Performance analysis: Service discovery

To evaluate the efficiency of the proposed approach to service discovery, we measured the number of messages exchanged while discovering services. To verify the effectiveness of the proposed method, it was compared with the approaches based on flooding or DHT. Specifically, we measured the number of messages required for publishing, deleting, and discovering services for the discovery architectures.

Figure 11 shows the number of the messages exchanged when nodes move randomly without specific moving patterns. We found that the proposed method needs fewer messages than DHT or flooding. This is because the proposed method dynamically selects a discovery architecture depending on the features of

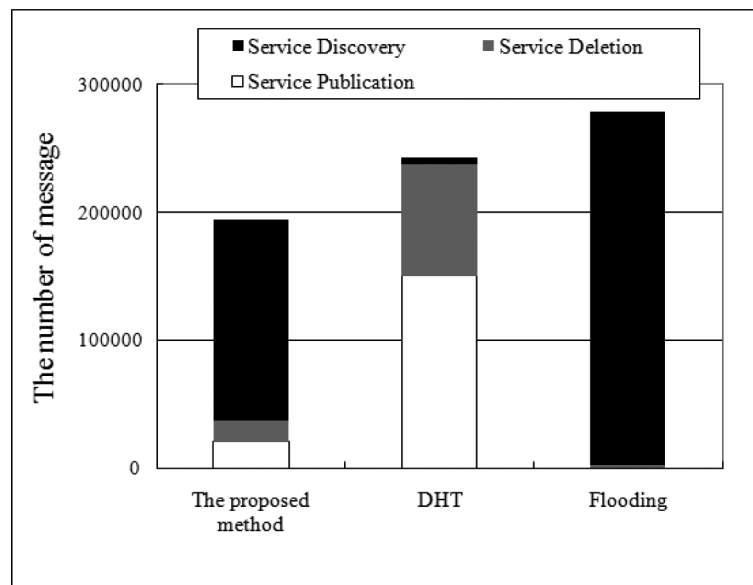


Fig. 11. The number of messages (RWP).

clusters unlike the conventional methods, which uniformly apply a discovery architecture. Particularly, since nodes move randomly in RWP, the proposed method selected flooding most often as a discovery architecture. resulting in a great part of the traffic while discovering services. Meanwhile, in the case of DHT, the large amount of messages was needed during cluster reconstruction, while the number of service discovery messages was small.

Figure 12 shows the amount of messages in the case of ELM. Flooding needed the largest number of messages, which were required for discovering services. In the case of DHT, while the number of service discovery messages was small, more messages were needed to reconstruct clusters by the movement of nodes. In contrast, the proposed method needed the smallest number of messages. Particularly, as a discovery architecture was selected based on the feature of clusters, the number of messages for service discovery was relatively small. Meanwhile, in the case of Flooding, since clusters were being comprised of more nodes, compared with the clusters in RWP, the number of the query messages broadcasted became larger.

6. Conclusions and future works

Web services are gaining momentum as standard technologies for implementing service-oriented computing systems in open environments. Recently, there is a growing interest in applying Web services to MANET in which the network topology of mobile devices is dynamically changing. However, most of the previous works do not sufficiently consider the features of MANET. They also use limited features such as the connectivity between nodes, and the capability and link duration of nodes. Additionally, they uniformly apply a single discovery architecture regardless of the dynamic nature of the mobile environment.

This paper constructs stable clusters by analyzing the mobility of nodes and selects an appropriate discovery architecture according to the characteristics of clusters. To analyze the mobility of nodes, the

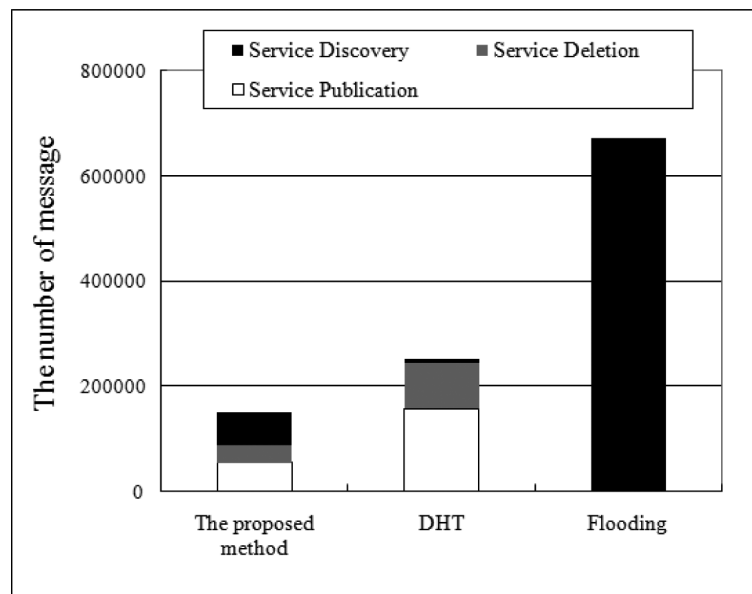


Fig. 12. The number of messages (ELM).

proposed method considers the moving direction and velocity of nodes, their sustained duration, and the connectivity with neighboring nodes. Based on this mobility analysis, the proposed method computes the suitability of the node as a cluster head and selects a node with the highest suitability as a cluster head.

Furthermore, it selects an appropriate service discovery scheme depending on the characteristics of clusters in an effort to reduce the number of the messages transmitted to manage clusters. Experimental results show that the proposed clustering method forms more stable clusters than previous works in terms of the number of the clusters created and deleted particularly in an environment, where nodes move with a certain moving pattern. However, the proposed method may be ineffective in an environment, where nodes are free to move, because the proposed mobility evaluation cannot be applied. Meanwhile, in terms of service discovery, the proposed method is more efficient than previous ones since it selects an appropriate discovery architecture depending on a cluster's characteristics unlike the conventional methods, which uniformly apply a discovery architecture. In the near future, we have a plan to develop a more sophisticated approach to performing a mobility analysis and an efficient method of discovering semantically accurate services quickly on resource-constrained mobile devices.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0025286).

References

- [1] A.A. Abbasi and M. Younis, A survey on clustering algorithms for wireless sensor networks, *Computer Communications* 30(14–15) (2007), 2826–2841.

- [2] A.M. Hanashi, I. Awan and M. Woodward, Performance evaluation with different mobility models for dynamic probabilistic flooding in MANETs, *Mobile Information Systems* **5** (2009), 65–80.
- [3] A. Obaid, A. Khir, H. Mili and L. Laforest. A routing based service discovery protocol for ad hoc networks, in: *ICNS'07: Proceeding of the 3rd Int'l Conf. Networking and Services*, IEEE Computer Society, Athens, Greece, 2007, p. 108.
- [4] B.D. Martino, Semantic web services discovery based on structural ontology matching, *International Journal of Web and Grid Services* **5**(1) (2009), 46–65.
- [5] B. Williams and T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, in: *MobiHoc'02: Proceeding of the 3rd ACM Int'l Symposium on Mobile Ad Hoc Networking & Computing*, ACM SIGMOBILE, Lausanne, Switzerland, 2002, pp. 194–205.
- [6] C.-C. Yang and Y.-C. Chang, A stability-based clustering technique and routing protocol for mobile ad hoc networks, *Information Science and Engineering* **24**(2) (2008), 469–481.
- [7] D.B. Johnson and D.A. Maltz, Dynamic source routing in ad hoc wireless networks, *Mobile Computing* **353**(5) (1996), 153–181.
- [8] D. Chakraborty, A. Joshi, T. Finin and Y. Yesha, GSD: a novel group-based service discovery protocol for MANETS, in: *MWCN'02: Proceeding of the 4th Int'l Workshop on Mobile and Wireless Communications Network*, IEEE Communications Society, Stockholm, Sweden, 2002, pp. 140–144.
- [9] D. Charkaborty, A. Joshi, Y. Yesha and T. Finin, Toward distributed service discovery in pervasive computing environments, *IEEE Transactions on Mobile Computing* **5**(2) (2006), 97–112.
- [10] D.C. Engelhart, A. Sivasubramaniam, C.L. Barrett, M.V. Morin, J.P. Smith and M. Morin, A spatial analysis of mobility models: application to wireless ad hoc network simulation, in: *ANSS'04 : Proceeding of the 37th Annual Simulation Symposium*, The Society for Modeling and Simulation International(SCS), Arlington, Virginia, USA, 2004, pp. 35–42.
- [11] D.L. McGuinness and F.V. Harmelen, Web Ontology Language (OWL), W3C Recommendation February 10, 2004, from <http://www.w3.org/TR/owl-features/>.
- [12] D. Taniar and J. Goh, On Mining Movement Pattern from Mobile Users, *International Journal of Distributed Sensor Networks* **3**(1) (2007), 69–86.
- [13] F.D. Tolba, D. Magoni and P. Lorenz, Connectivity, energy and mobility driven clustering algorithm for mobile ad hoc networks, in: *GLOBECOM'07: Proceeding of the IEEE Global Telecommunications Conf.*, IEEE Computer Society, Washington, DC, USA, 2007, pp. 2786–2790.
- [14] H. Artail, H. Safa, H. Hamze and K. Mershad, A cluster-based service discovery model for mobile ad hoc networks, in: *WiMob'07: Proceeding of the 3rd IEEE Int'l Conf. Wireless and Mobile Computing, Networking and Communications*, IEEE Computer Society, New York, USA, 2007, p. 57.
- [15] I. Stoica, R. Morris, D.L. Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek and H. Balakrishnan, Chord: A scalable peer-to-peer lookup protocol for internet applications, *IEEE/ACM Transactions on Networking* **11**(1) (2003), 17–32.
- [16] J.-C. Liang, J.-C. Chen and T. Zhang, Mobile Service Discovery Protocol (MSDP) for mobile ad-hoc networks, in: *ISADS'07: Proceeding of the 8th Int'l Symposium on Autonomous Decentralized Systems*, IEEE Computer Society, Sedona, Arizona, USA, 2007, pp. 352–362.
- [17] J. Haillet and F. Guidec, A protocol for content-based communication in disconnected mobile ad hoc networks, *Mobile Information Systems* **6** (2010), 123–154.
- [18] J. Xie, L.G. Quesada and Y. Jiang, A threshold-based hybrid routing protocol for MANET, in: *ISWCS'07: Proceeding of the 4th Int'l Symposium on Wireless Communication Systems*, IEEE Communication Society, Trondheim, Norway, 2007, pp. 622–626.
- [19] K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, A survey and comparison of Peer-to-Peer overlay network schemes, *IEEE Communications Surveys & Tutorials* **7**(2) (2005), 72–93.
- [20] R.S. Marin-Perianu, J. Scholten, P.J.M. Havinga and P.H. Hartel, Cluster-based service discovery for heterogeneous wireless sensor networks, *Parallel, Emergent and Distributed Systems* **23**(4) (2008), 325–346.
- [21] R.T. Meier, J. Dunkel, Y. Kakuda and T. Ohta, Mobile agents for service discovery in ad hoc network, in: *AINA'08: Proceeding of the 22nd Int'l Conf. Advanced Information Networking and Applications*, IEEE Computer Society, GinoWan, Okinawa, Japan, 2008, pp. 114–121.
- [22] S.N. Srirama, M. Jarke and W. Prinz, Mobile Web Services Provisioning, in: *AICT/ICIW'06: Proceedings of the Advanced Int'l Conf. Telecommunications and Int'l Conf. Internet and Web Application and Services*, IEEE Computer Society, Guadeloupe, France, 2006, pp. 120–126.
- [23] S.N. Srirama, M. Jarke and W. Prinz, Mobile web service discovery in peer to peer networks, in: *UMICS'07: Proceeding of the Workshop on Ubiquitous Mobile Information and Collaboration Systems*, SINTEF, Trondheim, Norway, 2007, pp. 531–542.
- [24] S.N. Srirama, M. Jarke and W. Prinz, Scalable mobile web service discovery in peer to peer networks, in: *ICIW'08: Proceeding of the 3rd Int'l Conf. Internet and Web Applications and Services*, IEEE Computer Society, Athens, Greece, 2008, pp. 668–674.

- [25] S. Wang, Q. Sun and F. Yang, Towards Web Service selection based on QoS estimation, *International Journal of Web and Grid Services* **6**(4) (2010), 424–443.
- [26] T.-H. Lin, H.-C. Chao and I. Woungang, An enhanced MPR-based solution for flooding of broadcast messages in OLSR wireless ad hoc networks, *Mobile Information Systems* **6** (2010), 249–257.
- [27] W.D. Maurer and T.G. Lewis, Hash table methods, *ACM Computing Surveys* **7**(1) (1975), 5–19.
- [28] W. Gao, Towards scalable and robust service discovery in ubiquitous computing environments via multi-hop clustering, in: *MOBIQUITOUS'07: Proceeding of the 4th Annual Int'l Conf. Mobile and Ubiquitous Systems: Computing, Networking & Services*, IEEE Computer Society, Philadelphia, USA, 2007, pp. 1–8.
- [29] Y. Yang, H. Hassanein and A. Mawji, A new approach to service discovery in wireless mobile ad hoc networks, *International Journal of Sensor Networks* **2**(1/2) (2006), 135–145.
- [30] Z.E. Bazzal, M. Kadoch, B.L. Agba, F. Gagnon and M. Bennani, A flexible weight based clustering algorithm in mobile ad hoc networks, in: *ICSNC'06: Proceeding of the Int'l Conf. Systems and Networks Communications*, IEEE Computer Society, Papeete, Tahiti, French Polynesia, 2006, p. 50.
- [31] Z. Gao, L. Wang, X. Yang and D. Wen, PCPGSD: An enhanced GSD service discovery protocol for MANETs, *Computer Communications* **29**(12) (2006), 2433–2445.

Yeon-Seok Kim received the B.S. degree in Information Engineering from MyongJi University, YongIn, Korea, in 2003 and the M.S. degree in Computer Science from Yonsei University, Seoul, Korea, in 2005. Currently, he is working toward the Ph.D. degree in Computer Science at Yonsei University. His research interests include internet computing, service-oriented computing, and mobile Web services. He is a member of the Korean Institute of Information Scientists and Engineers.

Yoo-Seok Shim received his B.S. and M.S. degrees in Computer Science from Yonsei University, Seoul, Korea, in 2007 and 2009. Currently, he is working as a researcher at LG electronics. His research interests include service-oriented computing and mobile Web services. He is a member of the Korean Institute of Information Scientists and Engineers.

Kyong-Ho Lee received his B.S., M.S., and Ph.D. degrees in Computer Science from Yonsei University, Seoul, Korea, in 1995, 1997, and 2001, respectively. Previously, he worked as a guest researcher in the IT Laboratories at NIST (National Institute of Standards and Technology), Maryland. He was a visiting scholar at University of California, Irvine in 2008. Currently, he is an associate professor in the Department of Computer Science at Yonsei University. His research interests include service-oriented computing, mobile Web services, and semantic Web. He is a member of the editorial board of *International Journal of Web Science*, *Journal of Information Processing Systems*, and *Journal of Korea Multimedia Society*.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

