

An adaptive low-overhead resource discovery protocol for mobile ad-hoc networks

Jui-Chi Liang · Jyh-Cheng Chen · Tao Zhang

Published online: 11 November 2010
© Springer Science+Business Media, LLC 2010

Abstract In this paper we propose a Distributed Hash Table (DHT)-based peer-to-peer resource discovery protocol designed for large-scale Mobile Ad-hoc NETWORK (MANET). The proposed Mobile Resource Discovery Protocol (MRDP) uses a distributed dynamic clustering algorithm to adaptively and rapidly group nodes in a MANET for resource discovery purpose, and utilizes DHTs to efficiently cache resource information in a peer-to-peer manner. We provide a probabilistic lower bound on the performance of the MRDP. We also present simulation results under different scenarios to show that MRDP has low message overheads and its performance is highly insensitive to the size of the MANET. These characteristics make MRDP suitable for large-scale MANETs.

Keywords Service discovery · Mobile Ad-hoc NETWORK (MANET) · Peer-to-peer networks

1 Introduction

In the past two decades, we have seen great progresses in mobile ad-hoc network (MANET) technologies.

The research community and industries have been exploring ways to apply MANET techniques to a growing range of applications in addition to battlefield networks, such as search/rescue missions and vehicle-to-vehicle communications. In order to provide value to users, it is desirable to have platforms or mechanisms that enable information sharing among network nodes, which enable distributed applications and services to be efficiently constructed or deployed on top of MANETs. In wired networks, such demands can be achieved by utilizing search engines or peer-to-peer platforms. However, the inherent heterogeneity of MANET has posed great challenges in designing resource discovery and sharing mechanisms. We identify the fundamental problems that need to be tackled when designing a resource discovery protocol for MANETs as: (1) how to distribute resources and services information to stimulate the discovery process, (2) how to discover resources and services in an efficient manner, and (3) how to retrieve the resources or services that have been found. Examples of resources in a MANET include files of common interests, environmental data sensed by sensors, and road traffic condition in vehicular networks.

One traditional way of doing resource discovery in MANETs is through broadcasting/flooding. Many studies have focused on improving the performance of broadcasting in MANETs. However, the excessive message overhead of broadcasting would degrade the network performance tremendously. Recently, Peer-to-Peer (P2P) technologies have been widely used in the Internet for distributed applications. One major advancement in P2P research is the introduce of Distributed Hash Tables (DHTs). DHTs have been proved to be highly efficient, robust, and scalable in message routing and locating objects [1–4]. The P2P information retrieval techniques share several important characteristics with that of doing resource discovery in MANETs, such as the ability

J.-C. Liang
Institute of Communications Engineering,
National Tsing Hua University, Hsinchu, Taiwan

J.-C. Chen (✉)
Department of Computer Science,
National Chiao Tung University, Hsinchu, Taiwan
e-mail: jcc@cs.nctu.edu.tw

T. Zhang
Applied Research, Telcordia Technologies,
Piscataway, NJ 08854, USA

to support unstable network links and unexpected node arrivals and departures, making them attractive to be deployed in MANETs. However, recent research studies have also shown that deploying P2P networks directly on top of MANET routing protocols would induce heavy message overhead [5, 6]. This is because P2P networks were originally designed as overlay networks for wired networks. Two neighboring peers in a P2P overlay network may be many physical hops apart in the underlying physical network, which is the main cause of excessive message overhead in MANETs. Thus, enhancements are needed to make P2P mechanisms work efficiently in MANETs. Cross-layer designs have been proposed to combat the routing inconsistency [6, 7]. By utilizing routing information provided by underlying routing protocols, the DHT-based resource discovery protocols can construct routing tables that better fit the actual physical topology.

In this paper, we propose a DHT-based peer-to-peer resource discovery protocol that is suitable for MANETs. Potential applications of the proposed protocol include information retrieval in sensor networks, intelligent gathering in battlefield ad-hoc networks, road condition gathering in vehicular networks, and file sharing over future 4G networks. The proposed protocol is designed to meet the following key requirements:

- *Low message overhead* Because nodes in a MANET communicate with each other wirelessly, message overhead, if not appropriately controlled, can greatly degrade the network performance. Hence, the resource discovery protocol for MANETs should have low message overheads, i.e. low number of messages that are needed to locate resources or services.
- *Scalability and robustness* The node topology of a MANET can change dynamically, and sometimes frequently, due to node mobility or energy drain-out. The size of a MANET can also vary widely, e.g. from tens to tens of hundreds of nodes. Therefore, the service discovery protocol for MANETs should have stable performance in a dynamically changing network. It should also be scalable and robust.

Based on above considerations, we propose Mobile Resource Discovery Protocol (MRDP), a peer-to-peer resource discovery protocol specially designed for MANET environment. Nodes that run the MRDP will form dynamic clusters for resource discovery. Resource information can be cached in each cluster by using DHTs. Unlike other cross-layer approaches, we design a resource discovery protocol that fully utilizes the scalability and robustness of DHT, yet minimizes the message overhead through adaptive caching. In addition, we provide a probabilistic lower bound on the performance of MRDP and

show that the protocol behaves consistently regardless of the network size. Our performance analysis and simulations show that MRDP outperforms broadcasting/flooding approaches and cross-layered DHT approaches. The MRDP is also scalable and robust in various MANET sizes. The contributions of this paper include:

- The proposed protocol forms the nodes in a MANET into dynamic clusters, and utilizes DHTs to cache and search for resources. The idea of clustering is not new, nor does deploying DHTs in MANET. However, to the best of our knowledge, designs and evaluations of efficient resource discovery mechanisms which utilize both clustering and DHTs have not been discussed in the literature before.
- We provide a probabilistic lower bound on the performance of the proposed resource caching mechanism. This lower bound serves as the theoretical foundation of the proposed protocol.
- We evaluate the performance of the proposed MRDP by extensive simulations. We compare with Ekta [6] and Virtual Ring Routing [7]. The proposed MRDP outperforms them in most scenarios.

An earlier version of this work was presented in [8], which showed preliminary simulation results. In this paper, we provide theoretical foundations for the proposed approach and enhance the design. Detailed simulation results are also given to further verify the performance of the proposed approach. The rest of this paper is organized as follows. Section 2 overviews related work. Section 3 presents the proposed MRDP. Section 4 provides performance comparison with other approaches. Section 5 concludes this paper.

2 Related work

Several industrial consortiums and organizations have made effort in standardizing resource discovery protocols. Examples include Universal Plug and Play (UPnP) [9], Jini [10], Service Location Protocol (SLP) [11], and Bluetooth Service Discovery Protocol (SDP) [12]. However, these protocols, except Bluetooth Service Discovery Protocol, assume a stable network. Therefore, they are not effective in handling the dynamic and frequent network changes in MANETs.

Several studies on resource discovery in ad-hoc networks have been presented in the literature. Because of the inherent nature of ad-hoc networks, most of the approaches are decentralized. They generally can be categorized into: (1) layered approach, and (2) cross-layered approach. Many of them share similar characteristics with P2P networks.

Layered approaches construct substrates somewhere between network layer and application layer to provide resource discovery capabilities. Message exchanges of resource discovery are done without the knowledge of the underlying physical connections and network topology, which would result in sub-optimal message forwarding paths [5, 6]. In [13], a P2P file sharing protocol called *ORION* searches files by broadcasting messages, and making nodes on the searching path cache information about the files for MANETs. Group-based Service Discovery (GSD) presented in [14] uses DARPA Agent Markup Language (DAML) to describe service information. It also utilizes the hierarchical semantic structures of DAML to create different groups of service information. Due to the sub-optimal forwarding path caused by layered approach, these mechanisms tend to have larger message overheads.

Cross-layered approaches, on the other hand, have been proposed in order to combat the sub-optimal forwarding problem of layered approach [6, 15, 16]. They generally integrate the resource discovery with network layer. By knowing the routing paths discovered by the network layer, the resource discovery substrate can deliver its message using near-optimal paths provided by the network layer, which results in fewer messages being exchanged in the network. In [15], Cheng proposed to integrate service discovery messages into the On-Demand Multicast Routing Protocol (ODMRP) to reduce the service query overheads. In [16], Kozat et al. proposed to integrate resource discovery protocols into network-layer routing protocols. In our literature survey, we found that the work done by Pucha et al. [6] has the closest goal as ours. In [6], the authors proposed Ekta, a DHT mechanism which integrates with the routing layer. However, we take a different approach by dynamically organizing the MANET into groups and perform DHT operations in each groups. We argue that our approach is more suitable for large scale MANETs in that it can reduce message overhead even more efficiently.

In addition to being a growing way of file sharing over the Internet, P2P networking technologies also provide promising means for resource sharing and dissemination over MANETs. The use of DHT-like protocols to provide resource discovery in ad-hoc networks has also received increasing attention [6, 7, 17, 18]. Structured P2P networks [1–4], which are fully decentralized, have been introduced to construct scalable P2P networks. With the help of DHTs, nodes in structured P2P networks can have a definite guide to route their query packets and maintain routing table states. The use of DHTs has made structured P2P scalable and robust. P2P networks share many similarities with MANETs. However, several recent studies have indicated that adopting the existing P2P mechanisms directly into

MANETs would cause excessive message overhead [5, 6], which will significantly degrade the performance of a MANET. The greatest challenge of adopting DHT-like protocols into MANETs lies in how to reduce message overhead efficiently. Thus, proper design of using DHT-like protocols in MANETs is crucial.

3 Proposed mobile resource discovery protocol (MRDP)

The resource discovery protocol proposed in this paper has two unique features:

- (1) It can group nodes in a MANET dynamically and in a decentralized manner into *information clusters* for the purpose of caching resource information. Dividing a large MANET into several smaller regions can effectively reduce the message overhead caused by resource discovery messages. The proposed dynamic formation cluster algorithm can be adjusted adaptively to fit different goals and environment.
- (2) The proposed protocol utilizes DHT to efficiently cache resource information in each information cluster. In wired peer-to-peer networks, the DHT provides a way to route messages in the overlay network. However, as aforementioned discussion, such an approach would cause excessive message overhead in a MANET. In our proposed protocol, thus, the DHT will not be served for routing purposes. Instead, the DHT is used to help caching resource information in clusters so that wide-area broadcasting could be avoided. The use of DHT not only further reduces the message overhead, but also provides a definite guide which tells nodes where to search and cache their resource information.

The combination of these two mechanisms leads to a highly scalable and robust solution which makes the proposed protocol suitable for large-scale MANETs.

3.1 System model and problem definition

Considering a MANET with N nodes in a two-dimensional space \mathcal{A} , each node has a maximum transmission range of r_0 . Without loss of generality, let $\mathcal{A} = [0, 1]^2$ and $0 < r_0 \leq 1$. Nodes are assumed to be uniformly distributed in the space. The MANET can be modeled as a random geometric graph $Ge(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_0, v_1, \dots, v_N\}$ is the set of all randomly deployed nodes and \mathcal{E} is the set of random edges. An edge (i, j) exists between two nodes v_i and v_j if and only if the *Euclidean distance* between v_i and v_j is less than or equal to r_0 . Each node has an Unique

Identifier (UID) to represent its identity. UIDs assigned to nodes are uniformly distributed in a m -bit key space. In practice, nodes in a network are identified by their IP or MAC addresses. Thus, one-way hash functions such as MD5 (128 bit) (IETF RFC 1321) and SHA-1 (160 bit) (IETF RFC 3174) can be used to map the nodes' IP or MAC addresses into m -bit integers and generate UIDs. Let $\mathcal{S}_p \subseteq V$ be the *resource providers*, which are nodes capable of providing one or more resources to others. Resources can be files, network information, environmental information, or hardware devices such as printers. Let $\mathcal{S}_r \subseteq V$ be the *resource retrievers* that may search and use the resources. We refer to nodes by using their UIDs throughout this paper. The nodes in \mathcal{S}_r will try to obtain the information of the resources provided by the nodes in \mathcal{S}_p .

We categorize the resources in a MANET based on two criteria:

- *Content Type* The content of a resource can be *time-variant* or *time-invariant*. The content of a resource is time-variant if it will change over time. On the other hand, the content of a resource is time-invariant if it is consistent over time.
- *Resource Availability* The availability of a resource can be *volatile* or *nonvolatile*. A resource is said to have nonvolatile availability if its existence is guaranteed. If the availability of a resource is uncertain, it is said to have volatile availability.

One typical example of *time-variant* resources is the environmental data sensed by sensors. In this case, the resource is *nonvolatile* if the sensors are immobile and data are gathered periodically. Shared files, on the other hand, are examples of *time-invariant* resources if content does not change over time. However, shared files may have *volatile* availability because their providers may join and leave the network arbitrarily. Problems arise when nodes try to locate time-variant and volatile resources. Mechanisms such as caching and replication may be used to accelerate the dissemination and discovery of time-invariant resources even when their providers are temporarily unavailable. However, for resources that are both time-variant and volatile, it is problematic to determine its availability in the network as well as its content correctness. Thus, in this paper we mainly consider resources that have time-invariant contents with either volatile or nonvolatile availability. The ability to discover and distribute time-invariant resource can be useful in scenarios such as a group of people sharing documents in a conference, fetching medical and communication resource information in a MANET built for disaster relief, or acquiring nearby roadside information in vehicular ad-hoc networks.

The goal of the proposed resource discovery protocol is to efficiently help users find the resources they need in a MANET. For nonvolatile resources, the goal is to locate them with very high probability and precision. If the resources are volatile, we want those resources to be found easily with high popularity. We also want to discard those resources with low demands to save memory space. The proposed MRDP comprises two major parts: (1) *Dynamic Information Cluster Formation* and (2) *Resource Discovery and Caching*. The Dynamic Information Cluster Formation organizes nodes in a MANET into groups according to some pre-defined criteria for the purpose of caching and replicate resource-related information. The Resource Discovery and Caching mechanisms utilize the organized structures to discover resources and to cache discovered resource information in a P2P manner using DHTs. Before elaborating on how they work, we first present the theoretical foundation of the design.

3.2 Theoretical foundation of the design

Because our goal is to design an efficient resource discovery protocol for MANETs through resource caching, the expected performance of the caching mechanism is a main concern. Hence, in this section we provide a probabilistic lower bound on the performance of the caching mechanism, which serves as the theoretical foundation of the proposed Mobile Resource Discovery Protocol.

3.2.1 Number of nodes in clusters

One thing we would like to know particularly is the average size of information clusters in the MANET. Here we mainly concern with the size of clusters in terms of the number of nodes in the clusters. The size and the number of clusters will have impact on several performance issues, including response time and control message overhead. Note that the cluster structure will be influenced by the cluster formation protocol. Therefore, the size of clusters vary in different scenarios. Due to the variations of cluster formation protocols and the random behavior of MANET topology, it is not possible to derive a closed-form analysis for the cluster size. However, by noting that the size of clusters is dominated mainly by the node degree, we can obtain a probabilistic bound for the cluster size.

We assume every node has a same transmission range r_0 . We only consider cases of $0 < r_0 < 0.5$ in the unit square \mathcal{A} because we are concerning with large-scale MANETs. As depicted in Fig. 1, the whole area can be divided into nine sub-areas according to the value of r_0 .

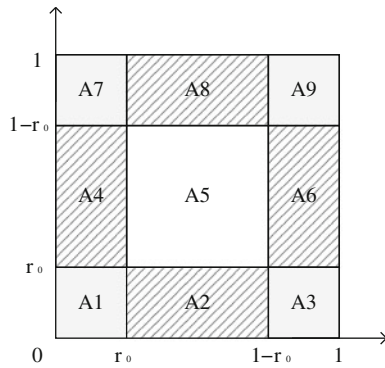


Fig. 1 Illustration of boundary conditions

The probability for a random node n_i to be of degree k in a MANET with m nodes can be obtained as follows:

$$Pr[\text{Degree}(n_i) = k] = Pr[n_i \text{ has } k \text{ neighbors}] = \sum_{j=1}^9 Pr[n_i \text{ has } k \text{ neighbors} | n_i \text{ in } A_j] Pr[n_i \text{ in } A_j] \quad (1)$$

Boundary conditions need to be tackled in deriving the probability distribution. As shown in Fig. 1, A_5 in the center does not suffer from any boundary limitation. $A_2, A_4, A_6,$ and A_8 reside on the sides of the boundary. Note that they are symmetric in terms of the boundary conditions. Similarly, $A_1, A_3, A_7,$ and A_9 reside in the corner and they are also symmetric in terms of the boundary conditions. Hence, (1) can be expressed as:

$$Pr[\text{Degree}(n_i) = k] = Pr[n_i \text{ has } k \text{ neighbors} | n_i \text{ in } A_5] Pr[n_i \text{ in } A_5] + 4 \cdot Pr[n_i \text{ has } k \text{ neighbors} | n_i \text{ in } A_4] Pr[n_i \text{ in } A_4] + 4 \cdot Pr[n_i \text{ has } k \text{ neighbors} | n_i \text{ in } A_1] Pr[n_i \text{ in } A_1] \quad (2)$$

Let the probability for a random node n_j to be a neighbor of another random node n_i , given that n_i is in $A_5, A_4,$ and $A_1,$ be $P_\alpha(x, y), P_\beta(x, y),$ and $P_\gamma(x, y),$ respectively. Because we consider uniformly distributed nodes in unit square and r_0 is normalized, we get:

$$P_\alpha(x, y) = \pi r_0^2, \quad \forall (x, y) \in A_5 \quad (3)$$

$$P_\beta(x, y) = r_0^2 \left[\pi - \text{acos}\left(\frac{x}{r_0}\right) + x \cdot r_0 \sin \text{acos}\left(\frac{x}{r_0}\right) \right], \quad \forall (x, y) \in A_4 \quad (4)$$

$$P_\gamma(x, y) = r_0^2 \left[\frac{\frac{3}{2}\pi - \text{acos}\left(\frac{x}{r_0}\right) - \text{acos}\left(\frac{y}{r_0}\right)}{2} \right] + xy + \frac{x \cdot r_0 \sin(\text{acos}\left(\frac{x}{r_0}\right))}{2} + \frac{y \cdot r_0 \sin(\text{acos}\left(\frac{y}{r_0}\right))}{2}, \quad \forall (x, y) \in A_1 \quad (5)$$

Substituting (3), (4), (5) into (2), we can derive the node degree distribution:

$$Pr[\text{Degree}(i) = k] = \binom{m-1}{k} \left[\iint_{A_5} P_\alpha(x, y)^k (1 - P_\alpha(x, y))^{m-1-k} dx dy + 4 \cdot \iint_{A_4} P_\beta(x, y)^k (1 - P_\beta(x, y))^{m-1-k} dx dy + 4 \cdot \iint_{A_1} P_\gamma(x, y)^k (1 - P_\gamma(x, y))^{m-1-k} dx dy \right] \quad (6)$$

Figure 2 shows the cumulative density function of the degree distribution. The analysis is verified by simulation. r_0 is set to 0.25 and the number of nodes are 50, 100, and 150. The expected node degree for 50, 100, and 150 nodes are 7.68, 15.52, and 23.35, respectively. Note that the actual cluster sizes may not be identically distributed as the expected node degree. For instance, if we let the nodes with greatest node degree in its vicinity to be the cluster heads in order to maximize the cluster size, the expected cluster size would be greater than the expected node degree of uniformly distributed nodes. The node degree distribution derived here gives us a probabilistic lower bound of the protocol performance. In our resource discovery protocol, the main concern is the cluster size.

3.2.2 Cache hit probability

The effectiveness of the MRDP will rely heavily on the cache mechanism. The cache mechanism is an efficient way to reduce the control overhead induced in the resource discovery process. If the desired resource information can be obtained in the local cluster that the requesting mobile belongs to, we can design MRDP with only three control messages: one for the query request, one for the query relayed by the cluster head through the Consistent Hashing operation, and one for the query reply.

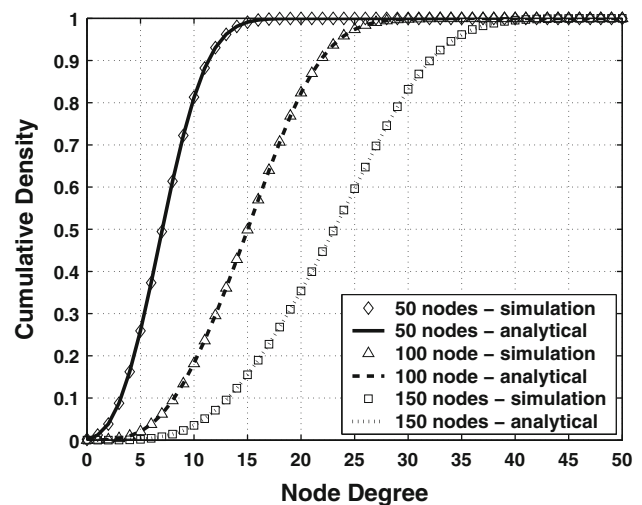


Fig. 2 Cumulative distribution function of node degree. (Transmission range is set to 0.25.)

The resources that are on high demand may be queried more frequently, thus staying in the cache space for a longer time. To this end, we adopt *soft state* to cache resource information. By using soft state, we can make popular resources to have higher cache hit probability. Also, those resources that are not in popular demand will be erased from the cache space to ease the processing and memory burdens of mobile nodes. Because resource information is cached in soft state, the main factor that will affect the cache hit probability is the popularity of the resource information.

Let X denote the inter-arrival time of resource requests of each user. We assume that X follows exponential distribution with mean μ . Let τ be the duration that mobile nodes will hold resource information before erasing them. Also let N be the random variable denoting the cluster size. We get:

$$Pr[N = n] = Pr[\text{Degree}(\text{head}) = n - 1], \quad n \geq 1 \quad (7)$$

The probability of cache hit can be obtained as:

$$Pr[\text{A resource queried can be satisfied by cache}] = Pr[X \leq \tau] = 1 - e^{-\frac{\tau}{\mu}} \quad (8)$$

The value of τ/μ is the ratio between expiration period of resource information and mean inter-arrival time of resource requests. We refer to it as the EI Ratio. Figure 3 shows the cache hit probability for different cluster sizes. EI Ratios of 0.1, 0.333, and 0.5 are depicted. In clusters with 5 nodes, the cache hit probability is 0.4 when EI Ratio is 0.1. This means we can fulfill 40% of resource requests through cached information with an expiration period of 1/10 of the mean resource request inter-arrival time. The cache hit probability becomes higher as the size of clusters and EI ratio increase. In the 5-node cluster case, it is over

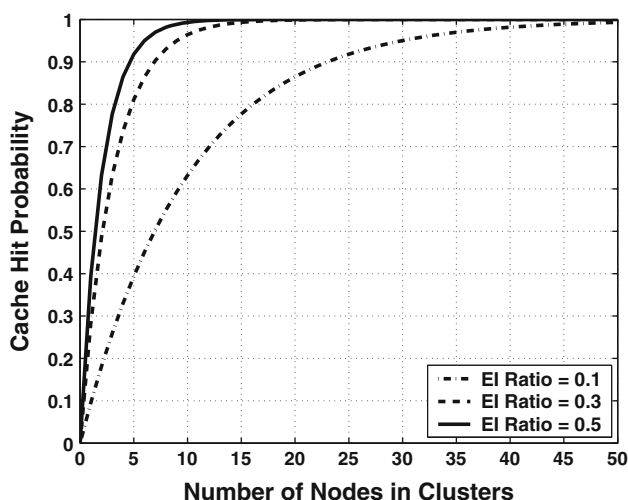


Fig. 3 Cache hit probability for different cluster sizes

0.8 for EI Ratio of 0.333 and over 0.9 for EI Ratio of 0.5, which means a large portion of resource requests can be fulfilled by the cached information in the cluster.

Taking a MANET with 100 nodes and $r_0 = 0.25$ as an example, from Fig. 2 we can see that with high probability (over 0.95) a node will have degree greater than 5. It implies the probability of forming a cluster with more than 5 nodes is at least 0.95. If we can tune the expiration period to get an EI Ratio of roughly 0.1 for a specific resource, we can expect that, with high probability, the cache hit ratio is over 40% for this resource.

Having the probabilistic lower bound on the caching performance as our design basis, we now move on to the protocol design of the proposed MRDP.

3.3 Dynamic information cluster formation

Based on the theoretical foundation provided in Sect. 3.2, we propose algorithms to dynamically group mobile nodes in a MANET. Instead of proposing a specific cluster formation algorithm and hoping it would result in good performance under various situations, we adopt a more generalized approach. Specifically, we do not set restrictions on how the clusters should be formed. We only require that there exist a geometric relation between cluster heads and members. That is, cluster heads and members are n -hop neighbors and form a connected component. Cluster heads are selected based on some *grades* that are measured by each node. The measurement of grades can be network-dependent. For instance, in a MANET with unstable links, the grade can be a measurement of link quality. Nodes with best link quality will become the cluster heads in order to have more stabilized clusters. In a MANET with low node mobility, the grade can be node degree. As a result, the nodes with the most number of neighbors will become the cluster heads and larger clusters can be constructed. Novel metrics such as *information density* [19] can be used as grades to facilitate the resource discovery process as well.

The proposed cluster formation algorithm is similar to the cluster formation algorithm in [20]. We evolve the algorithm to adapt to the needs for resource discovery. In the process of forming dynamic information clusters, every node can be in one of the following states: *Unknown*, *Member*, or *Head*. The radius of the cluster is RAD_c hops from the cluster head. Each node also has a countdown timer whose maximum value is $TIMER_MAX$ seconds, and broadcast a Hello Message (HM) every $HELLO_INTERVAL$ seconds with Time-To-Live (TTL) field set to one. The HM contains the UID, the state of the sender, and the network-dependent grades that is used to determine whether the nodes should become cluster heads. It also contains all of the neighbor information as well as the cluster information that the sender

knows, including nodes within RAD_c hops and cluster heads within $(2 \times RAD_c + 1)$ hops. The range for cluster head information is larger because the maximum distance between the cluster heads of two clusters of radius RAD_c is $(2 \times RAD_c + 1)$. This allows cluster heads to know how and where to contact their neighbor clusters. Sending rate of the HM can be adaptively changed according to the mobility pattern of the nodes. In a highly mobile environment, the sending frequency of the HM can be higher in order to obtain accurate node information. On the other hand, if nodes barely move, the sending frequency of the HM can be tuned lower to save network resources such as bandwidth and node energy. Like other cross-layer approaches, the HM can also be integrated with network-layer or link-layer beacon messages.

Algorithm 1 shows how a node handles incoming HMs. Upon hearing a HM, a node will store the UID and the state of each node the HM carries in its *Neighbor Table*. The grades of each node are also recorded. The node then checks the state and grade of each entry, and performs different procedures depending on the state of itself. When an *Unknown* node learns that there is a *Head* node within RAD_c hops, it changes its state to *Member* to join the cluster represented by this cluster head. If a *Head* node finds that there is another *Head* node with a better grade within RAD_c hops, it starts its countdown timer. Finally, if an *Unknown* node n_i knows the existence of another *Unknown* node n_j within RAD_c hops and n_j has a better grade than n_i , n_i will restart its countdown timer from $TIMER_MAX$.

Algorithm 1 Upon Hello Message Reception

```

1: procedure RECVHELLOhm
2:   for each neighbor entry  $e$  in  $hm$ 
3:     AddEntry( $e.addr$ ,  $e.hop\_dist$ ,  $e.state$ ,  $e.grade$ )
4:     if  $e.state = Head$  and  $e.hop\_dist < RAD_c$  then
5:       if  $me.state = Unknown$  then
6:          $me.state \leftarrow Member$ 
7:       else if  $me.state = Head$  and  $e.grade$  is better than  $me.grade$  then
8:         StartTimer()
9:       end if
10:    else if  $e.state = Unknown$  then
11:      if  $me.state = Unknown$  and  $e.grade$  is better than  $me.grade$  then
12:        RestartTimer()
13:      end if
14:    end if
15:  end for
16: end procedure

```

Algorithm 2 When Timer Expires

```

1: procedure TIMEREXPIRE
2:   if  $me.state = Unknown$  then
3:      $me.state \leftarrow Head$ 
4:   else if  $me.state = Member$  then
5:     if no Head exists within  $RAD_c$  hops then
6:        $me.state \leftarrow Unknown$ 
7:     end if
8:   else ( $me.state = Head$ )
9:     if the Head with a better grade still exists within  $RAD_c$  hops then
10:       $me.state \leftarrow Member$ 
11:    end if
12:  end if
13: end procedure

```

Algorithm 2 shows the operations a node executes when its countdown timer expires. Nodes in different states have different process to do. A node in *Unknown* state changes its state to *Head*. This implies that it has the best grade in its vicinity. A *Member* node checks whether it is still within RAD_c hops of any cluster head. If not, it changes its state back to *Unknown* and attempts to join or form another cluster. A *Head* node checks whether the other *Head* node that caused its timer to start still exists within RAD_c hops. If so, it changes its state to *Member* and becomes a cluster member. This operation can avoid *Head* nodes to change their states when other *Head* nodes move into their coverage temporarily. Thus, it stabilizes the cluster structure.

Figure 4 demonstrates the cluster formation process. To simplify our illustration, we use UIDs as the grades to form clusters. We assume that all nodes are initially in *Unknown* state, that is, no clusters have been formed yet. Note that the design of the cluster formation algorithm can also work correctly even when some *Member* and *Head* nodes exist in the MANet already. The UID key space is 4-bit in length (i.e. [0,15]) and RAD_c is set to one. All nodes are in *Unknown* state in the beginning as shown in Fig. 4(a). They exchange HMs and learn the existence of their neighbors. The timers of the nodes with the best grades (smallest node IDs) in their vicinity will expire first, which makes these nodes change their states to *Head* (Fig. 4(b)). When the *Unknown* nodes receive HMs from the *Head* nodes and learn that they are under the coverage of at least one *Head* node, they change their states to *Member* and become cluster members (Fig. 4(c)). It is also possible that a node does not possess the best grade in its vicinity nor is it under coverage of any *Head* nodes (Node 7 in Fig. 4(c), for example). After the timer of this node expires, it will also change its state to *Head* (Fig. 4(d)). Eventually, all nodes are either in *Head* or *Member* state and clusters are formed (Fig. 4(e)).

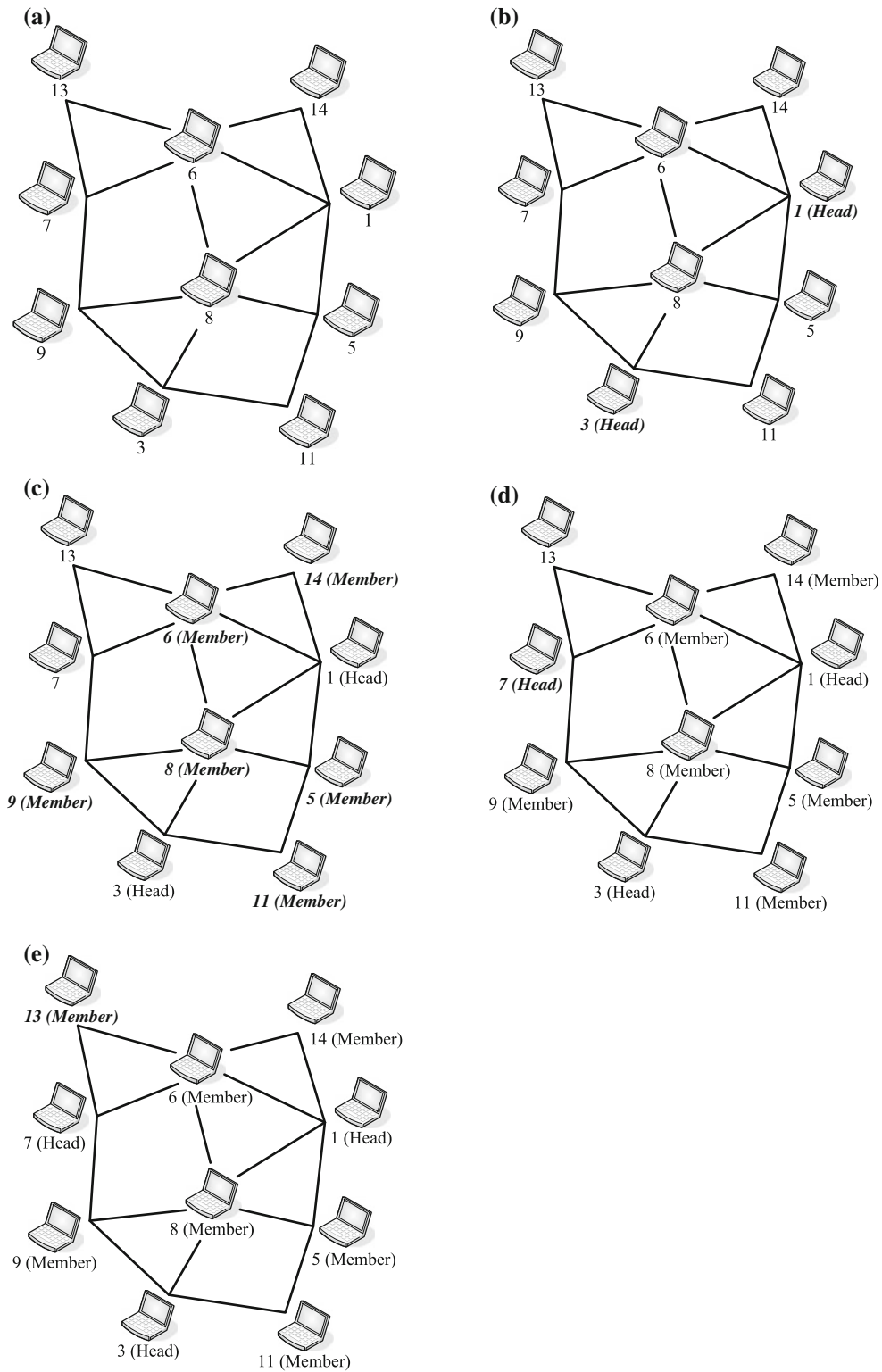


Fig. 4 Dynamic cluster formation

3.4 Consistent hashing

To facilitate the resource discovery process, each information cluster utilizes the consistent hash function [21] to

cache and search resource information among the nodes in the cluster. Note that other DHT algorithms [2–4] are also applicable. Consistent hashing is chosen here because it is simple to implement. Every *Head* node treats all the

Member nodes within RAD_c hops as its cluster members, and uses their UIDs to form a cluster-local key space. The one-way hash function that is used to generate UIDs for nodes is now used to map resource information or description strings to integers of the same length of m -bit as UIDs. These m -bit integers will be used in the resource discovery and caching process. We refer to the m -bit integers generated by the one-way hash function as the *information keys*.

The consistent hashing function takes an information key as input and maps it to one of the UIDs in the cluster-local key space. The consistent hashing first sorts the UIDs to be mapped in increasing order and form a circular linked-list. That is, the last element of the sorted list links to the first element. Assuming there are n UIDs sorted in increasing order and the key space is m -bit in length, the consistent hash function maps the input information key k to the *nearest* UID i in the key space using modulo operations. In consistent hashing, i is defined as the smallest UID that is greater than or equal to $(k \bmod 2^m)$. If $(k \bmod 2^m)$ is greater than the largest UID of the n UIDs, i is defined as the smallest UID. The consistent hash function has a nice property that when UIDs of the cluster-local key space changes, only a small fraction of information keys need to be re-mapped. This property is crucial for MANETs where topology changes happen frequently. Let's take the upper left cluster of Fig. 4(e) as an example. Figure 5 depicts the operations of the consistent hash function. Assuming a resource R_1 is found and its information key is 10. To insert R_1 into this cluster-local key space, the consistent hash function will map it to node 14

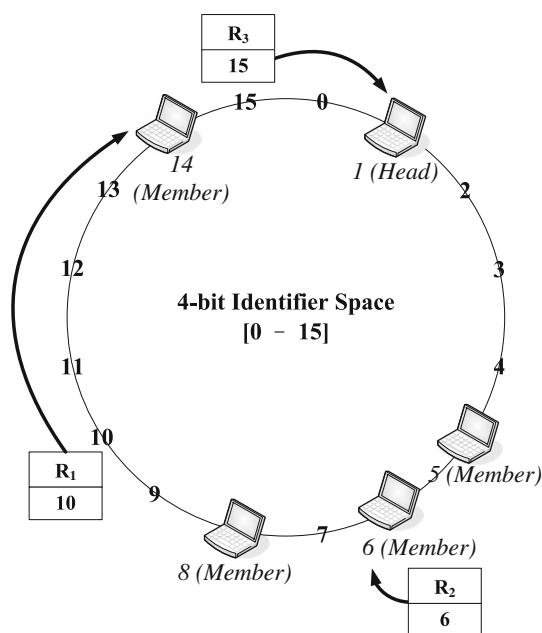


Fig. 5 Illustration of the operations of the consistent hash function

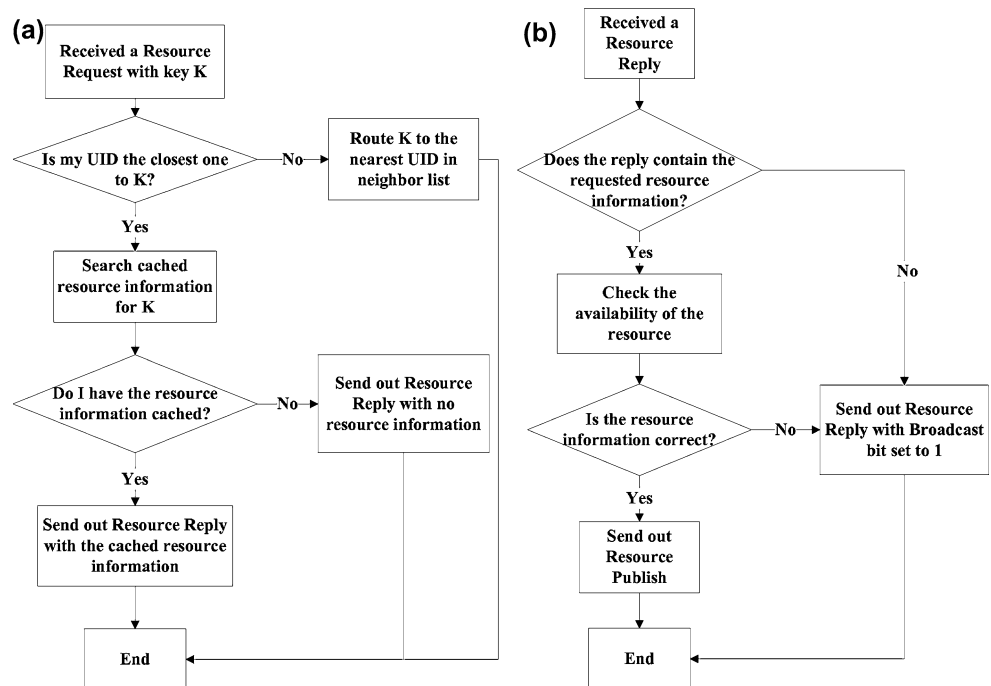
because node 14 is the first node that has an UID greater than or equal to $(10 \bmod 2^4)$. Similarly, following the same procedure, resource R_2 and R_3 will be mapped to node 6 and node 1, respectively.

3.5 Resource discovery and caching

To achieve the probabilistic lower bound derived in Sect. 3.2, we propose protocols for resource discovery and caching based on the dynamic cluster formation and consistent hashing algorithms discussed above. Initially, all nodes know nothing about each other. The only way to perform resource discovery is through broadcasting/flooding. Thus, here we first assume that some sort of resource discovery has been made and the nodes have partial knowledge on the resource information. Figure 6 shows the flow charts of the proposed MRDP. When a user wants to find a specific resource, the user packs the resource description along with the information key used to represent the description in a *Resource Request* packet. The user then sends the packet to the head of the cluster the user belongs to. When a *Head* receives the Resource Request, it uses consistent hashing to map the information key in the packet to one of its *Member* nodes. It then relays the Resource Request to the mapped node to see whether it has cached the requested resource information before. Finally, when the mapped *Member* node receives the Resource Request, it checks its own resource database. If the requested resource information is found, it sends back a *Resource Reply* containing the resource information. Otherwise, it sends back a Resource Reply without resource information, indicating that no resource information is found and the required resource is not cached in the cluster.

If the user receives a Resource Reply which indicates that no resource information is found in the local cluster, it sends out another Resource Request which is then broadcast to all nodes. Each node that receives the Resource Request will check whether it has the requested resource information. Eventually, the user will receive a Resource Reply if the requested resource exists in the MANET. After assuring the availability and correctness of the requested resource, the user sends out a *Resource Publish* packet to the user's cluster head, which contains the resource information and the information key of the resource information. Similar to Resource Request, the *Head* uses consistent hashing to map the resource information to a *Member* node and relays the packet to that *Member*. The *Member* that receives the Resource Publish then caches the resource information in its database for a period of time. Later, when other users in the cluster wish to access the same resource, their requests may be fulfilled locally and the wide-area broadcasting can be avoided.

Fig. 6 Flow charts of MRDP: **a** flow when a node receives a Resource Request. **b** flow when a node receives a Resource Reply



As discussed in Sect. 3.2, the resource information is cached by using soft state. When a resource information is not accessed for EXPIRE_PERIOD seconds, it will be removed from the cache. A popular resource will be updated frequently. Thus, it can stay in the cache for a long period of time. There are two advantages of using soft state to store resource information. First, resource providers do not need to periodically update their resource information to the whole MANET. The updates of resource information are automatically done by users that access the resource. Thus, less network resources are consumed. Secondly, frequently used resources will stay in the cache for a long period of time, and resources that are not popular will be removed quickly. This leads to more efficient use of nodes' storage space. Because we are concerning large scale MANETs, the number of resource information could be large. Caching the resource information permanently would consume too much memory and energy for mobile nodes.

4 Performance evaluation

This section presents the performance comparison of the proposed MRDP with other approaches. First, we would like to know how the proposed MRDP performs comparing with the basic flooding/broadcast approach. Moreover, it is of our interests that whether the proposed MRDP can be more efficient than pure DHT-based approaches. In our literature survey, we found that Ekta [6] and VRR [7] are two mechanisms that share similar goals as ours. Ekta is a

resource discovery mechanism that is based on cross-layer DHTs. VRR is a DHT-based routing protocol that is built upon the link layer. Thus, resource discovery using *physical layer flooding*, *layered DHT approach*, *Ekta*, and *VRR* are chosen as comparison bases. The comparison is done by using ns-2. The parameters used in our simulations are listed in Table 1.

In MRDP, we use the SHA-1 to generate 160-bit integer UIDs for mobile nodes from their IP addresses. Information keys are also generated by the SHA-1. For demonstration purpose, we choose the UIDs as the grades to form clusters. The nodes with smaller UIDs in their vicinity will become cluster heads. Because many ad-hoc routing protocols utilize periodical beacon messages to assure inter-node connectivity, we can incorporate our HM into these

Table 1 Default parameters used in simulation

Transmission range	250 (m)
Carrier sensing range	550 (m)
Mobility model	Semi-Markov Smooth
Moving speed	2–5 (m/s)
Mean pause time	60 (s)
Propagation model	Two ray ground
MAC protocol	802.11b
Ad-hoc routing protocol	AODV
Simulation area	1,000 x 1,000 (m)
HELLO_INTERVAL(AODV)	2 (s)
TIMER_MAX	10 (s)
EXPIRE_PERIOD	90 (s)

beacon messages to reduce control overhead. In our simulations, we embed the HM in AODV's periodical beacon messages. For the layered DHT and cross-layered DHT approaches, we also embed necessary node information in AODV's beacon messages in order to provide a fair comparison. The DHT we choose is Chord [1], which is a DHT implementation based on consistent hashing. For the layered approach, Chord is implemented on top of the AODV routing protocol. We utilize its DHT to perform resource advertisement and discovery. For the cross-layered approach, we utilize node distance information provided by AODV and select DHT routing table entries for Chord based on the physical distances between nodes. In Chord, to shorten the DHT routing path, the whole node ID space is divided into several segments. The protocol stores one node in each segment as routing entries which are called *fingers*. Thus, for the cross-layered approach, when selecting fingers for the Chord DHT, it takes into account not only proper prefix of node ID, but also physical distance.

In simulation, the movement of nodes follows the Semi-Markov Smooth (SMS) model [22]. In the SMS model, node movement follows four consecutive phases: Speed Up, Middle Smooth, Slow Down, and Pause. The SMS model is chosen because it is a better realization of node mobility, comparing with conventional mobility models such as Random Waypoint or Random Direction. In the SMS model, given that initial node positions and moving directions are independently uniformly distributed on $[0, 1)^2 \times [0, 2\pi)$, the location and moving direction of nodes remain uniformly distributed all the time [22].

4.1 Performance metrics

4.1.1 Reply-to-query ratio (RQR)

The RQR represents the probability that a user correctly receives a Resource Reply in response to a Resource Request the user sent. Let N_q be the number of queries sent, and N_r be the number of correct replies received. The RQR is measured as $RQR = \frac{N_r}{N_q}$ on a per user basis.

4.1.2 Cache-to-reply ratio (CRR)

The CRR represents the probability that a resource reply comes from a local cache in the local cluster that the requester belongs. Higher CRR means less packets are transmitted during the resource discovery process. Thus less network resources are consumed. Let N_c be the number of replies obtained from local cache. The CRR is measured as $CRR = \frac{N_c}{N_r}$ on a per user basis.

4.1.3 Query response time (QRT)

The QRT is measured as the time difference between the transmission of the first request packet and the reception of the first correct reply packet.

4.1.4 Message overhead

We define message overhead as the number of control packets generated per resource request by the resource discovery protocol during the resource discovery process.

4.2 Impact of cluster radius

Bigger cluster radius potentially will reduce the message overhead of MRDP. In the extreme case where the cluster radius is so large that there is only one cluster head in the MANET, the cluster head will behave like a centralized server. Under such condition, the control message overhead would be reduced because the single cluster head has all the node information in the MANET. Therefore, resource discovery and caching messages can be delivered to the most appropriate node in a direct manner. However, having only one cluster head creates several drawbacks. This centralized approach does not fit into the dynamic nature of MANET. It results in performance bottleneck. It also creates a single point of failure. Besides, the failure of the cluster head will bring down the entire MANET.

Figure 7 shows the simulation results of the RQR as a function of RAD_c . The degradation of the RQR when RAD_c grows results from the aforementioned drawbacks of having large cluster radius. As depicted in the figure, smaller cluster radius is more favorable in terms of RQR. Smaller cluster radius is also more robust because the failure of a specific cluster head only affects a small portion of the nodes in the MANET. Therefore, we will emphasize on scenarios of clusters with 1-hop radius in the rest of the paper.

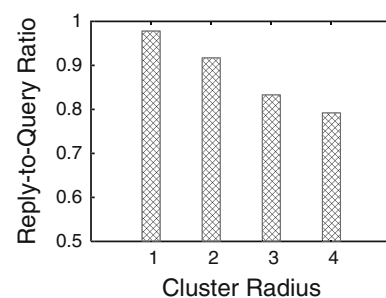


Fig. 7 RQR as a function of RAD_c

4.3 Impact of node mobility

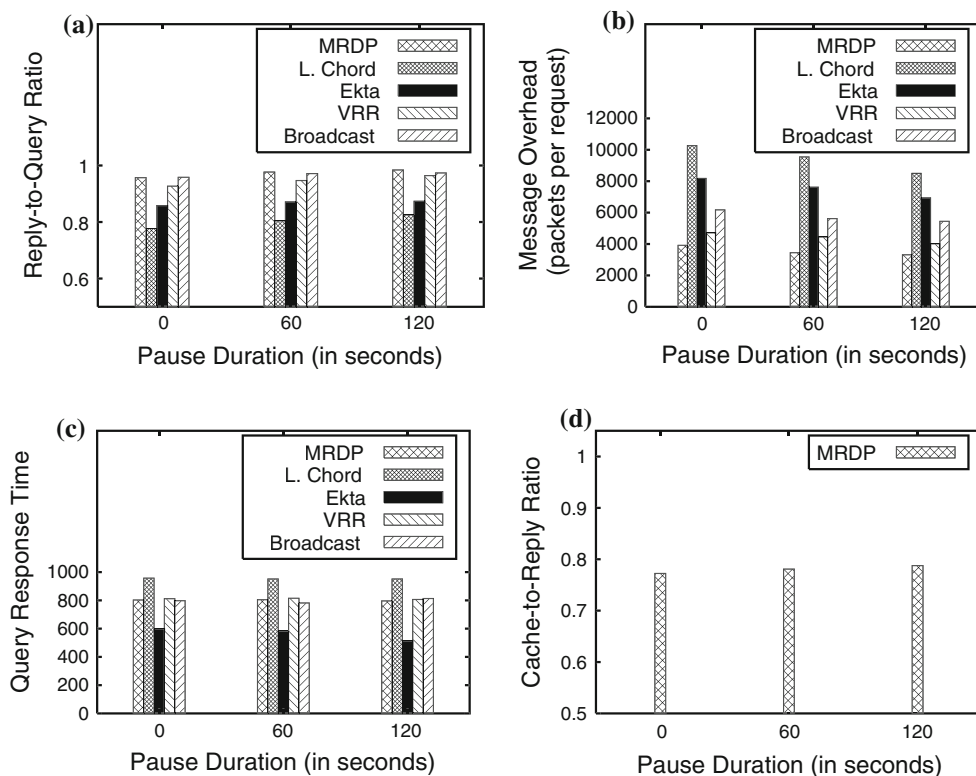
The five different approaches of resource discovery are compared under different level of node mobility. Mean node pause times of 0, 60, and 120 are selected to represent high, medium, and low node mobility, respectively. We generate 10 different traces of the SMS mobility model, each with 100 nodes and 2–5 m/s of moving speed. The other simulation parameters are set as that shown in Table 1. In each run of the simulation, one resource is generated randomly on one node. Other nodes send queries for the resource with exponentially distributed mean inter-arrival time of 180 s. Simulations of each type of mobility are carried out on each of the 10 mobility traces. We record the results after all the protocol structures are stationary. The results discussed below are averages of the 10 traces.

Figure 8(a) shows the RQR of the five approaches. Figure 8(b) shows the message overhead in terms of number of packets generated per resource request. For the five approaches, it is observed that the RQRs are negatively related to the message overhead. The RQRs of MRDP and Broadcast are above 0.95. VRR has slightly lower RQRs in high mobility scenario, and comparable RQRs in low mobility scenario. This degradation of RQRs of VRR in high mobility scenario is due to DHT path breaks. Layered chord generates excessive amount of messages and tremendously degrades the network performance. Thus, it has the lowest RQR. Ekta improves message overhead because

of its better forwarding path of requests. However, its message overhead is still higher than the overheads of MRDP, VRR and Broadcast approach.

We identify the message overhead generated by each protocols as two parts: the overhead of protocol maintenance and the overhead of message forwarding. The overhead of protocol maintenance includes overhead caused by cluster formation in MRDP, DHT construction and updates in Layered Chord and Ekta, and DHT routing table setup and repair in VRR. The Broadcast approach has no overhead of protocol maintenance. The overhead of message forwarding refers to all messages generated by resource queries, replies, and forwarding. Layered Chord and Ekta have large amount of message overhead mainly due to its protocol maintenance. Because they maintain network-wide of DHTs, vast number of messages needs to be generated periodically in order to ensure the correctness and accuracy of the DHTs. Although Ekta has lower overhead of message forwarding comparing with Layered Chord, its overall message overhead is still considered a huge burden for the network. On the other hand, the Broadcast approach has no overhead of protocol maintenance at all, but its overhead of message forwarding is the largest among the five approaches. VRR, as a routing protocol, has moderate overhead of protocol maintenance. Message forwarding does not cause much overhead in VRR since routing paths are maintained periodically. However, because routing in VRR is based on DHT rather

Fig. 8 Simulation results of MRDP, Broadcasting, Layered Chord, Ekta, and VRR under different node mobility



than physical distance, messages may follow sub-optimal routes. Thus, the average path length in VRR may be substantially longer than other MANET routing protocols such as AODV. The benefit of dynamic cluster and small-scale DHT can be demonstrated here. The combination of dynamic clusters and small-scale DHT can achieve a balance between the two types of message overhead, and efficiently reduce the impact to the network performance caused by message overhead. The node mobility has little effect on the performance of the five approaches. In general, lower node mobility improves the RQRs and decreases message overhead. However, the relative performance differences of the five approaches do not vary much.

Figure 8(c) illustrates the QRT of each approach. Layered Chord has the largest QRTs due to its imperfect forwarding path when doing resource request. Ekta has the best QRTs because it has optimized forwarding path. MRDP, VRR and Broadcast has comparable QRTs. The QRT of MRDP is a combination of two conditions. The first one is smaller, in which case a cache hit happens in the local cluster. The second one is when cache miss happens and network-wide broadcast is performed, which is just like Broadcast. The actual result of the QRT of MRDP is determined by the CRR. Figure 8(d) shows the CRRs of MRDP in each mobility scenario. With resource information expiration time being set to 90 s and mean inter-arrival time of resource requests being set to 180 s, the CRRs are around 0.78 in all three scenarios. This means

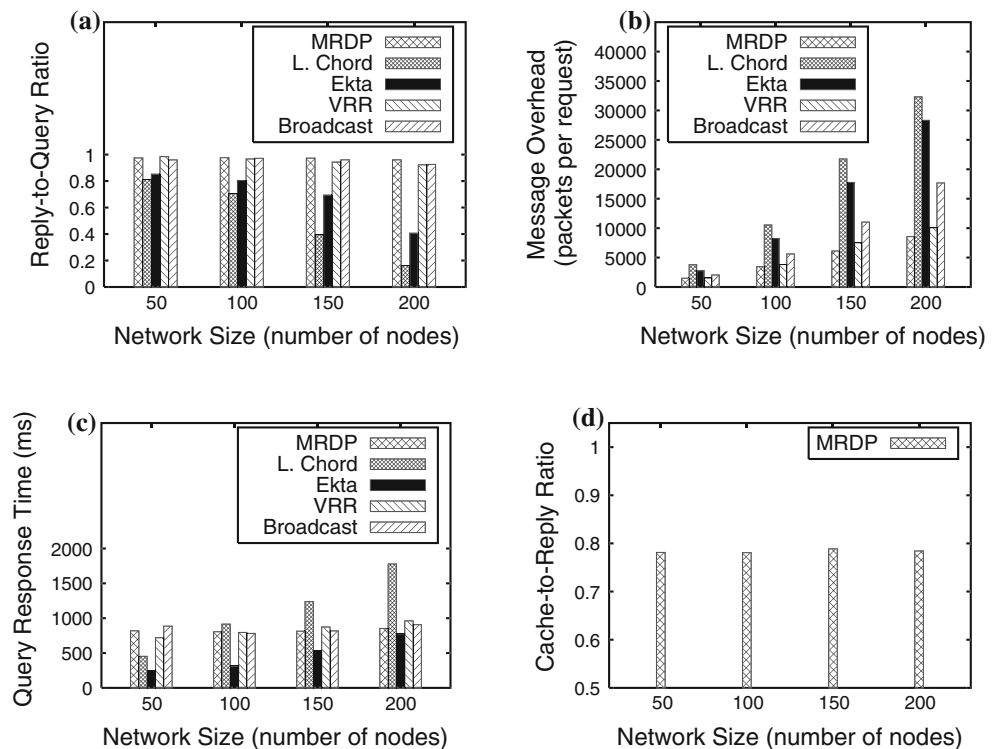
more than 70% of resource requests can be fulfilled by the cache in the clusters that the senders of the requests belong to. The CRR value determines not only the QRT, but also the overhead of message forwarding. In general, we would like the CRR value as high as possible to reduce both the QRT and message overhead.

4.4 Impact of network size

In Fig. 9, we fix the mean pause time of nodes at 60, and vary the number of nodes in the network. Four different sizes are evaluated: networks with 50, 100, 150, and 200 nodes. Similarly, we generate 10 traces of the SMS mobility model for each scenario. Rest of the simulation setup and parameters are the same as those in previous section.

Figure 9(a) shows the RQRs of the five approaches under different network sizes. Figure 9(b) shows the message overhead. The negative relation between the RQR and the message overhead is even more clear here. When the network size grows large, the message overhead requires to maintain the DHTs also grows. It can be observed that for the Layered Chord, the message overhead has greatly exceeded the network capacity in large networks. Besides, its RQR drops to an unacceptable value. Ekta has better performance than Layered Chord, but its RQR still drops largely when the network size grows large, mainly due to its excessive message overhead. Both Layered Chord and Ekta generate large number of messages due to their needs

Fig. 9 Simulation results of MRDP, Broadcasting, Layered Chord, Ekta, and VRR under different network sizes



for network-wide DHT maintenance. Their overhead of protocol maintenance in the 150-node and 200-node cases has grown beyond the network capacity and results in packet lost and RQR degradation. The message overhead of the Broadcast, though not as huge as Layered Chord and Ekta, also grows proportionally to the network size. Slight drops in RQR can be observed for the Broadcast in the 150-node and 200-node cases, which indicate the effect of increased message overhead. The message overhead of VRR is also proportional to the network size. However, the overhead caused by VRR does not exceed the capacity of the network. Hence, degradations in RQRs is not obvious for VRR. MRDP has the best performance in terms of RQR and message overhead. From the results, we can see that the overhead of protocol maintenance dominates the total message overhead of Layered and Ekta. When trying to utilize the advantages of DHT in MANETs, the first challenge is to reduce the message overhead. The advantage of MRDP can be seen here. The design of MRDP can efficiently reduce the amount of traffic generated and make it insensible to network size. This makes MRDP suitable to be deployed in large scale MANETs.

Figure 9(c) shows the QRT of the five approaches. Figure 9(d) shows the CRR of MRDP under different network size. Because Layered Chord and Ekta forward resource requests according to network-wide DHTs, the size of the network has far more influence on their QRTs. For Layered Chord, the QRT grows faster than MRDP, VRR and Broadcast when the network size grows large. For Ekta, the interworking between AODV and Chord successfully decreases the length of the forwarding path. Thus, even its QRT grows when network size becomes larger, it still has the smallest QRT. However, following the trend of growth, we believe there exist a point beyond which the QRT of Ekta exceeds that of MRDP, VRR and Broadcast. As a routing protocol, VRR has advantages in terms of QRT. However, as we mentioned earlier, VRR may construct sub-optimal routing paths because of its DHT-based routing table. Thus the QRT of VRR grows as network size increases. The growth is in an acceptable

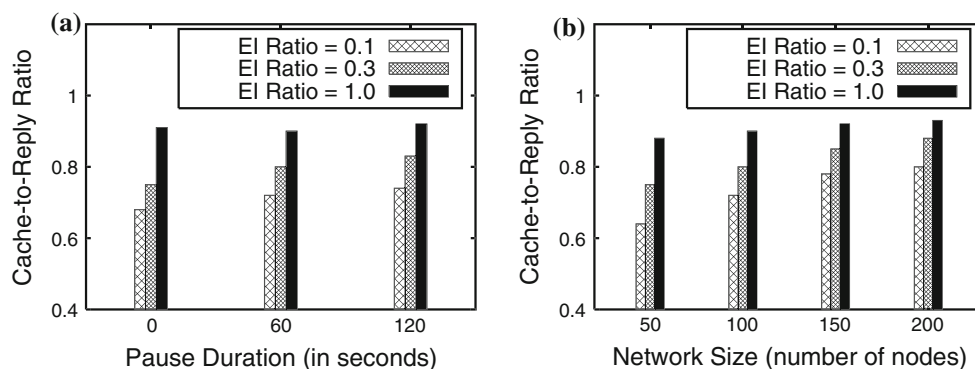
range, and does not influence its performance much. With CRR of around 0.78, the QRT of MRDP, VRR and Broadcast are comparable.

From the results, it is observed that Broadcast is the simplest and most efficient way to perform service discovery in MANET when the network size is small. However, as the network size grows, the message overhead caused by flooding would paralyze the network. VRR and MRDP perform especially well in large networks because of the use of node clustering and DHT-based routing. Generally speaking, VRR and MRDP have comparable performance in all scenarios. However, as a resource discovery mechanism, MRDP is easier to implement and deploy than VRR. To deploy VRR, a new routing module has to be inserted to replace the original routing protocol. Moreover, conversion mechanisms between existing routing protocol and VRR may be needed for old applications that are built upon existing routing protocols. On the other hand, MRDP can be deployed directly upon existing routing protocols. No modifications are needed to the system. MRDP can work as a standalone process that serves the needs for resource discovery.

4.5 Impact of resource request rate

Because the CRR has great impact on the performance of MRDP, we would like to know the behavior of the cache mechanism when the resource request rate varies. In Sect. 3.2 we have derived probabilistic bounds on the CRR. In this section, we vary the resource request rate and conduct simulations under different node mobility and network size to verify the probabilistic bounds we derived. The simulation settings are identical to the previous sections. We control the resource request rate in terms of the EI ratios defined in Sect. 3.2. Specifically, with EXPIRE_PERIOD fixed at 90 s, we control the mean inter-arrival time of resource request to simulate different rates. The mean inter-arrival times are exponentially distributed, and the values we select are 900, 270, and 90 s. Thus the EI ratios are 0.1, 0.3, and 1.0, respectively. Figure 10(a)

Fig. 10 Simulation Results of Cache-to-Reply Ratio with 0.1, 0.3, and 1.0 EI ratio under different node mobility and network sizes



shows the CRRs of each EI ratio under different node mobility. Figure 10(b) shows the CRRs of each EI ratio under different network sizes. Node mobility has little effect on the CRRs, while increases in network size. The simulated CRRs are slightly higher than the analytical bounds provided in Sect. 3.2. Because in our simulations we form clusters based on their node IDs, the results are close to our analysis above, which is based on uniformly distributed nodes. The simulation results show that even with small numbers of node and low EI ratio, the CRR is still above 0.5, which demonstrates the effectiveness of constructing cluster-local DHTs for caching resource information.

5 Conclusion

This paper presents the MRDP, a resource discovery protocol designed for large-scale MANETs. It is designed to support efficient resource discovery with low message overhead and stable performance over MANETs of various sizes. Inspired by the structured P2P technologies, we design a cache scheme that uses distributed hash tables to reduce the number of messages needed to locate a resource. Combining it with dynamic information clustering, high percentages of resource discovery queries can be resolved in local clusters. Thus it reduces the resource discovery overhead. We have derived a probabilistic lower bound for the performance of MRDP. The results indicate that with high probability, the proposed MRDP can achieve good performance in large scale MANETs. Also, simulation experiments verify that the proposed MRDP has steady performance in terms of resource information discovery and caching. It can also minimize the network resources consumption. Simulation results also indicate that VRR and MRDP have comparable performance in terms of resource discovery. However, MRDP is much easier to implement and deploy comparing to VRR, which is one of the main advantage of MRDP.

Acknowledgment This work was sponsored in part by National Science Council (NSC) under the grant number 98-2221-E-009-187-MY3.

References

1. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M. F., Dabek, F., & Balakrishnan, H. (2003). Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11, 17–23.
2. Ratnasamy, S., Francis, P., Handley, M., & Karp, R. (2001). A scalable content-addressable network. In *Proceedings of conference on applications, technologies, architectures, and protocols for computer communications*. San Diego, CA, USA, August 2001, pp. 161–172.
3. Zhao, B. Y., Huang, L., Stribling, J., Rhea, S. C., Joseph, A. D., & Kubiatowicz, J. D. (2004). Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22, 41–53.
4. Rowstron, A., & Druschel, P. (2001). Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *Proceedings of the 18th IFIP/ACM international conference on distributed systems platforms (Middleware)*. Heidelberg, Germany, November 2001, pp. 329–350.
5. Hsieh, H. Y., & Sivakumar, R. (2004). On using peer-to-peer communication in cellular wireless data networks. *IEEE Transactions on Mobile Computing* 3, 57–72.
6. Pucha, H., Das, S., & Hu, Y. (2004). Ekta: An efficient DHT substrate for distributed applications in mobile ad hoc networks. In *Proceedings of the sixth IEEE workshop on mobile computing systems and applications (WMCSA)*. English Lake District, UK, December 2004, pp. 163–173.
7. Caesar, M., Castro, M., Nightingale, E. B., O’Shea, G., & Rowstron, A. (2006). Virtual ring routing: Network routing inspired by DHTs. In *Proceedings of ACM SIGCOMM*. Pisa, Italy, September 2006, pp. 351–362.
8. Liang, J.-C., Chen, J.-C., & Zhang, T. (2007). Mobile service discovery protocol (MSDP) for mobile ad-hoc networks. In *Proceedings of IEEE International symposium on autonomous decentralized systems, special session on ad-hoc networks in automotive networking and applications (ISADS)*. Sedona, AZ, March 2007, pp. 352–362.
9. Microsoft Cooperation, “Understanding universal plug and play: a white paper,” <http://www.upnp.org/resources/whitepapers.asp>. June 2000.
10. Sun Microsystems, “Jini architecture specification,” <http://www.sun.com/software/jini/specs/jini1.2.html/jini-title.html>. December 2001.
11. Guttman, E., Perkins, C. E., Veizades, J., & Day, M. (1999). Service location protocol, version 2. IETF RFC 2608, June 1999.
12. Bluetooth Specification Part E (1999). Service discovery protocol (SDP). <http://www.bluetooth.com>. November 1999.
13. Klemm, A., Lindemann, C., & Waldhorst, O. P. (2003) A special-purpose peer-to-peer file sharing system for mobile ad hoc networks. In *Proceedings of IEEE vehicular technology conference*. Orlando, FL, USA, 2003, pp. 2758–2763.
14. Chakraborty, D., Joshi, A., Yesha, Y., & Finin, T. (2002). GSD: A novel group-based service discovery protocol for MANETs. In *Proceedings of the 4th international workshop on mobile and wireless communications network*. Stockholm, Sweden, September 2002, pp. 140–144.
15. Cheng, L. (2004). Service advertisement and discovery in mobile ad hoc networks. In *Proceedings of conference on computer supported cooperative work (CSCW), 2004*. [Online]. Available: <http://citeseer.ist.psu.edu/article/cheng02service.html>.
16. Kozat, U. C., & Tassiulas, L. (2003). Network layer support for service discovery in mobile ad hoc networks. In *Proceedings of IEEE INFOCOM*. San Francisco, CA, USA, April 2003, pp. 1965–1975.
17. Araujo, F., Rodrigues, L., Kaiser, J., Liu, C., & Mitidieri, C. (2005). CHR: A distributed hash table for wireless ad hoc networks. In *Proceedings of the 25th IEEE international conference on distributed computing systems workshops*. Columbus, OH, USA, June 2005, pp. 407–413.
18. Delmastro, F. (2005). From Pastry to CrossROAD: Cross-layer ring overlay for ad hoc networks. In *Proceedings of the third IEEE international conference on pervasive computing and communications workshops (PerCom)*. Kauai, HI, USA, March 2005, pp. 60–64.
19. Fiore, M., Casetti, C., & Chiasserini, C.-F. (2007). Efficient retrieval of user contents in MANETs. In *Proceedings of IEEE INFOCOM*. Anchorage, AK, USA, May 2007, pp. 10–18.

20. Amis, A. D., Prakash, R., Vuong, T. H. P., & Huynh, D. T. (2000). Max-min d-cluster formatin in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM*. Tel Aviv, Israel, March 2000, pp. 32–41.
21. Karger, D., Lehman, E., Leighton, F., Levine, M., Lewin, D., & Panigraphy, R. (1997). Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web. In *Proceedings of the 29th annual ACM symposium on theory of computing*. El Paso, TX, USA, May 1997, pp. 654–663.
22. Zhao, M., & Wang, W. (2006). A novel semi-markov smooth mobility model for mobile ad hoc networks. In *Proceedings of IEEE GLOBECOM*. San Francisco, CA, USA, November 2006, pp. 1–5.

Author Biographies



working, resource allocation, scheduling, and performance evaluation of wireless networks.

Jui-Chi Liang received his B.S. degree from the Department of Computer Science, National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2005. One year after his Master's study in the Institute of Communications Engineering, he was transferred into the Ph.D. program due to academic excellence. He is now a Ph.D. candidate. He was a summer intern at Telcordia Technologies, Piscataway, NJ, USA, in 2007. His research interests include ad-hoc net-



Tsing Hua University (NTHU), Hsinchu, Taiwan since 2001 as

Jyh-Cheng Chen received the Ph.D. degree from the State University of New York at Buffalo in 1998. Dr. Chen is a Professor in the Department of Computer Science, National Chiao Tung University (NCTU), Hsinchu, Taiwan. He was with Bellcore/Telcordia Technologies, Morristown, NJ, USA, 1998–2001, and Telcordia Technologies, Piscataway, NJ, USA, 2008–2010. He has also been with the Department of Computer Science, National

assistant/associate/full/adjunct professor. He is a coauthor of the book *IP-Based Next-Generation Wireless Networks*, (Wiley, 2004). He has published over 80 papers. He is the holder of 19 U.S. patents. Dr. Chen is a Senior Member of the IEEE, and a Senior Member of the ACM. He is a Technical Editor of the IEEE Wireless Communications. He was a guest editor of the IEEE Journal on Selected Areas in Communications, special issue on “All-IP Wireless Networks”, May 2004. He was the Technical Program Co-Chair of the Ninth IFIP/IEEE International Conference on Mobile and Wireless Communications Networks (MWCN '07) held in Ireland, September 2007. He was the Technical Program Co-Chair of the Third IEEE International Conference on Information Technology: Research and Education (ITRE '05). He has been on the technical program committee of numerous international conferences, including IEEE INFOCOM 2005–2006, IEEE GLOBECOM 2005–2009, and IEEE ICC 2007–2011. He was a Tutorial Speaker at IEEE GLOBECOM 2002, 2003, and 2006 in the subject of next-generation wireless networks.



Tao Zhang is Chief Scientist and Director of Wireless and Vehicular Networking at Telcordia Technologies, Piscataway, New Jersey, USA. He directs research and development on wireless networks, vehicular networks, and intelligent transportation systems. Dr. Zhang holds 19 U.S. patents on wireless and vehicular networks. He co-authored the book “IP-Based Next Generation Wireless Networks” published by John Wiley & Sons in 2004.

He is a member of the US National Academies' Strategic Highway Research Program 2 (SHRP-2) Technical Expert Task Group on Statistics, Models, and Methods. He serves on the Industry Advisory Board for the Michigan Center for Advancing Safe Transportation through the Lifespan (M-CASTL). He has been an editor for the Journal of Wireless Networks (WINET) since 2003, an associate editor for the IEEE Transactions on Vehicular Technology from 2008 to 2009, and a guest editor of the IEEE Journal on Selected Areas in Communications (JSAC) Special Issue on All-IP Wireless Networks in 2004. Dr. Zhang has also served as general chair, technical program committee chair, and in many other roles for a number of international conferences.