The Journal of Supercomputing (2020) 76:6809–6833
https://doi.org/10.1007/s11227-019-03139-3

A location and time-aware resource searching scheme in mobile P2P ad hoc networks

Shuai Zhou[1] · Xianfu Meng[2]

Published online: 6 January 2020

Abstract
With the advanced development of wireless communication techniques and the increasing application of smart devices, the mobile P2P ad hoc networks (i.e., P2P MANETs) are attracting more attention. P2P MANETs can be applied to the environments where the communication infrastructure is down due to natural disasters or political tensions. In such a network, how to improve resource search efficiency has been an important research focus. Most existing researches put emphasis on the location-based peers' clustering but paid less attention to the time factor to implement resource search approaches, resulting in low search efficiency. This paper first proposes a novel location-based peers' clustering mechanism and a time-aware partners' selection scheme. Then, we present a resource search algorithm employing both pull and push approaches based on the finding that peer movements are often repeated on a day-to-day basis to cope with peers' mobility issue in P2P MANETs. The simulation results show that our resource search scheme could both improve the successful search rate and reduce the propagated messages.

Keywords Mobile P2P ad hoc network (P2P MANET) · Resource search · Pull and push

✉ Xianfu Meng
xfmeng@dlut.edu.cn

Shuai Zhou
zhoushuai@neusoft.edu.cn

[1] School of Computer and Software, Dalian Neusoft University of Information, No. 8, Software Park Road, Dalian 116023, China

[2] School of Computer Science and Technology, Dalian University of Technology, No. 2, Linggong Road, Dalian 116024, China

1 Introduction

In recent years, peer-to-peer (P2P) networks have been widely applied in many areas, such as P2P messaging, P2P resource sharing and P2P VOD systems, due to their characteristics of openness, autonomy and anonymity [1]. Initially, P2P networks could be classified into the hybrid P2P networks and the pure P2P networks, depending on whether or not a centralized index server is required. However, with the increasing use of mobile devices, such as smart devices, the need for wireless or mobile P2P networks, especially for the mobile P2P ad hoc networks (i.e., P2P MANETs), is also increasing [2, 3]. A P2P MANET can be considered as the collection of autonomous mobile peers, which communicate with each other wirelessly without the help of any fixed network infrastructure and any centralized control [4]. All the peers in a P2P MANET form a dynamic and self-configuring topology. Each peer participating in a P2P MANET can play any of the three roles in a transaction: (a) a client requesting a service; (b) a server providing a service; (c) a router relaying a requesting message to other nodes. P2P MANET can be applied in residential quarters, campus and remote areas where the communication infrastructure has not been constructed or areas where the communication infrastructure is down due to natural disasters or political tensions [4–15]. As pointed out in [15], acquiring data (resource) in P2P MANET is time critical. For example, in the event of an earthquake, the communication infrastructure might be down. So, using the P2P MANET may be the unique way for the victims to communicate with the outside. In such situation, how to send the location information to the online neighbors and how to obtain the rescue information as soon as possible are crucial to the victims. Compared to wired P2P networks, P2P MANETs are faced with more problems to be dealt with, including the low data transmission efficiency, the frequent topology changes and the short radio transmission range [4–6]. In such environments, how to improve the efficiency of resource searching, which is one of the main functionalities of a P2P MANET, has been an important research focus [5]. To this end, a lot of strategies have been proposed, including both replication- and cluster-based [5, 6].

Replication-based resource searching strategies improve the search efficiency by replicating resources, e.g., files, to multiple peers. Considering the dynamicity of a P2P MANET, the replicas should be replicated to the relatively stable peers. For example, the work in [5] replicates resources by constructing a connected dominating set based on the network topology graph. Here, a connected dominating set refers to a unique path along the network where all peers either belong to the path or are at most one hop away. As pointed out in [6], however, such strategy could only be used in static networks. Since the mobility of peers in a P2P MANET is more frequent and more uncontrollable than any other types of P2P networks, maintaining a connected dominating set to increase resource availability in such a network is a difficult and complicated task [6], which would both waste much network traffic and incur high complexity.

Cluster-based resource lookup strategies aim at increasing resource search efficiency by clustering the peers whose locations are within a certain range [6–8].

For example, in work [7], peers are clustered based on locations (i.e., the peers' wireless transmission range, which is a radius covering the peers to which the current peer could make one-hop connection), and the peers with low mobility are selected as the cluster heads which are responsible for managing their clusters. Such strategy, however, can only be effective in networks with low mobility [4].

Existing work [9–11] indicated that in a mobile P2P network, peer movements are often repeated on a day-to-day basis. This means that even though being in the same location, the *partners* a peer is able to interact with may be largely different in different time intervals due to the peer's limited radio transmission range and strong mobility. For example, in a person's house, the *partners* the householder is able to interact with via a P2P MANET in daytime may be largely different from those at night. This gives us a hint that we can improve resource search efficiency in P2P MANET environments by using location and time-aware approach, rather than relying only on the location-based clustering method. Based on this consideration, this paper presents a location and time-aware resource searching scheme, *LoTiSearch* ("*LoTi*" is short for *Location* and *Time*-aware), in P2P MANETs. Our main contribution spans the following:

1. To tackle the issues of peers' mobility and short transmission range, we propose a location-based peers' clustering mechanism and a time-aware *partners*' selection scheme, under which a sender's request for a resource can be sent to the peers whose locations are closest to the sender and online probability is higher in the current time interval.
2. To improve resource search success rate, we present the pull and push-combined resource lookup algorithms, which make the best use of the created location-based clusters, the selected time-aware *partners* and the peers' mobility.
3. We present an algorithm used to incrementally make adjustment on the created clusters, so as to both adapt to the dynamic changes of network topology and reduce the clusters' maintenance overhead in P2P MANET environments.

The remainder of the paper is organized as follows. Section 2 presents the related work on the resource lookup mechanisms in P2P MANETs. In Sect. 3, we present several definitions and relevant information management mechanism, as well as the overview of the components involved in our scheme. The location-based clusters' creation and maintenance, as well as the time-aware *partner* selection approach are analyzed in Sects. 4 and 5. In Sect. 6, we detail the pull and push-combined resource search process for P2P MANETs, and Sect. 7 analyzes the complexity of our approach. Simulations and respective results are supplied in Sect. 8. Section 9 concludes the paper and gives our future research focus.

2 Related work

In recent years, rapid advancement has been made in wireless communication techniques. Accordingly, the application areas of wireless communication techniques and networks become more extended with the respective advancements. Cellular systems [12], wireless local area networks (WLAN) and wireless wide area networks (WWAN) [13], as well as mobile P2P ad hoc networks (P2P MANETs) [14] are examples of such application areas, in which P2P MANET represents the main research focus [14]. This is because a P2P MANET is able to be established anywhere and anytime in a self-organized form due to its characteristics of peers' mobility and self-configuring [4]. In a P2P MANET, each peer can only connect to its one-hop neighbor peers within the wireless transmission range. Any transaction between two peers which are two-hop away from each other needs routers to relay messages or services. In such a network, resource (service or message) sharing can be considered as one added-value functionality offered by the P2P MANET. Due to the peers' mobility and short wireless transmission range, how to improve resource search efficiency is a main research focus in P2P MANETs. To this end, researchers have proposed a lot of strategies, mainly including the replication-based strategies, the cluster-based strategies and other kinds of strategies.

(1) The replication-based strategies

Replication-based strategies improve resource search efficiency by replicating resources (e.g., files) to multiple peers. How to select the peers to which a resource is replicated is one of the key issues to be solved in such strategies. Atsan et al. [5] proposed a virtual backbone-based solution, in which the network nodes construct a connected dominating set based on the network topology graph. Data replication and lookup are implemented based on the connected dominating set, expecting to minimize the number of nodes in the network involved in resource lookup and replication process [5]. Pushpalatha et al. [15] also employed the dominating set to replicate data. This strategy is composed of two phases. The first phase is used to identify and minimize the dominating nodes, on which the data would be replicated. In the second phase, the strategy identifies a stable node, to which a replica is relocated from a server if the server moves out of range from its clients. To do this, the authors proposed several relevant algorithms, including the dynamic data replica allocation algorithm, the mobility prediction algorithm, the replica relocation algorithm with subgraph centrality principle and the algorithm for finding a subgraph centrality of a node. These algorithms are complementary to each other. For example, the mobility prediction algorithm should be performed before a decision on the replica relocation is made. However, since these algorithms are required to periodically monitor the connectivity of each peer's neighbors for relocating the replicas, the maintenance cost would be largely increased. Also, as mentioned in Sect. 1, maintaining a dominating set in the highly dynamic P2P MANETs is a difficult and traffic-consuming task [6].

Victer Paul et al. [16] presented a service cache management for mobile peer-to-peer (MP2P) networks to facilitate the efficient retrieval of services. Specifically, the nodes in a P2P MANET are made to form a distributed spanning tree which consists of several spanning trees. Each spanning tree has a head node which is responsible for caching the service information of the leaf nodes in its spanning tree. The ant colony algorithm is used to optimize the path between two head nodes or between a head node and its leaf nodes, so as to reduce the number of messages passed among the nodes. However, similar to the dominating set-based strategies, maintaining such distributed spanning tree is not an easy task. It would consume much network traffic and incur high complexity in the highly dynamic P2P MANET environments.

(2) The cluster-based strategies

Cluster-based strategies [7, 8, 17–19] improve search efficiency by clustering peers within the wireless transmission range. In [7], peers are clustered based on locations, and the cluster heads are selected by using the mobility parameter (speed). The nodes with lower mobility are candidates for cluster heads. Such strategy, however, can only be effective in networks with low mobility [4]. In work [8], each peer periodically broadcasts a beacon message. All the peers that received the beacon message within the wireless transmission range are considered as the members of the peer's cluster. When a peer wants to search a resource, it sends a request to all the peers in its cluster, i.e., using flooding approach. However, since the cluster is constructed by periodically broadcasting the beacon messages, it would consume much precious network traffic. Moreover, using flooding approach to search resources in P2P MANET would exacerbate the problem.

Wu [18] used the concept of stable groups to organize peers with stable connectivity. The stable group is defined based on the stable connectivity of peers, and the stable connectivity refers to the relative stability in distance or correlated mobility pattern over time. In other words, if two peers are moving together at a similar speed toward the same direction, which are evaluated by using the variation of their received signal strength, then the two peers can be put into the same group. However, in the real-world mobile P2P networks or P2P MANETs, such approach has limited application scenarios.

Wang et al. [19] cluster peers based on the peers' interests which are provided by the peers themselves. When a peer joins the system, it is asked to provide the information of its interest, and the peers with the same interest are put into a same group. However, there must be a centralized server (i.e., the base station) in the system, by which the peers' interests can be processed. Furthermore, how to classify peers' interests is another issue which should be considered carefully.

(3) Other strategies

The research in [20] took both the locations and the replicas as the factors to improve search efficiency in mobile P2P networks. In [20], the authors constructed a super peer-based architecture, under which they presented a replica allocation scheme for improving data availability in mobile P2P networks. A super peer is assumed to be the peer that generally does not move outside a given region and has maximum remaining battery power and processing capacity. Each peer periodically sends the read–write log of its managed resources and the read log of its managed replicas to its super peer, based on which the super peer can detect the peer's mobility patterns. Each super peer is responsible for exploiting peer mobility patterns to allocate replicas, so as to improve data availability. However, in a dynamic P2P MANET environment, how to select such a super peer and how to avoid the problem of single point of failure are the crucial issues.

Yang [21] proposed a location-based service discovery strategy with push approach in P2P MANETs. In the strategy, each peer periodically sends service advertisement to its one-hop neighbor peers within the wireless transmission range, and its neighbor peers forward the service advertisement to their neighbor peers. This process continues until all the peers received the advertisement. Since the service advertisement consists of the peer's ID and location as well as the relevant service information, any peer that received the service advertisement can get the service indicated by the service advertisement, if necessary. However, this strategy suffers from at least two shortcomings. First, the blind and periodical service advertisement pushing operations make almost all the peers in the network receive the same number of service advertisements, leading to high bandwidth consumption and traffic overhead. Second, how to cope with the trade-off between the frequency of sending service advertisements and the service discovery success rate is a difficult problem. This is because the more frequent the service advertisements are broadcast, the higher the service discovery success rate is and the higher the bandwidth is consumed, and vice versa.

The work in [22] proposed a location-aware service discovery protocol and a location-aware service selection strategy in P2P MANETs. In the location-aware service discovery protocol, if node j receives a request from node i, the protocol verifies the distance (d_{ij}) between peer i and peer j. If d_{ij} is greater than a threshold R_i, the request is discarded by peer j. In the location-aware service selection strategy, it takes the requester's geographic location as one of the factors to select a service provider from all the responders for the requester. However, as pointed out by the authors of work [22], such protocol and strategy cannot work well in sparse networks with frequent partitions.

Ajay et al. [23] implemented three resource discovery strategies in P2P MANET environments, including *Standard Flooding Based Resource Discovery Method*, *Standard Random Walk Based Resource Discovery Method* and *Standard Gossip Based Resource Discovery Method*. These methods were originally proposed and implemented in wired P2P networks. According to the simulation results, the traditional flooding method is better than the other two methods in terms of the network overhead, the success rate and the query response time. This outcome was achieved under the situation that the authors only used 1 *walker* in the random walk method, which is passed or forwarded to one randomly selected neighbor node. Due to the fact that the peers in a P2P MANET have strong mobility over time, only using 1 *walker* could not reflect the real ability of random walk method in resource searching process. In addition, the authors did not analyze why the network overhead of using flooding method is lower than that of using random walk method in detail.

Seddiki et al. [24] proposed a P2P lookup mechanism over MANETs based on popularity and proximity. This work first constructs a P2P overlay based on peers' physical proximity and then employs the resource popularity-biased random walk to send search requests for resources. A peer's proximity is achieved by broadcasting a connection request message, and a resource's popularity is calculated using the current peer's local knowledge about the resource and the global knowledge estimated by the current peer's direct neighbors. Note that the resource popularity is used to determine the number of walkers and their TTL (time to live) under the assumption that the more the resource is popular, the higher the probability to find it in a close area is. However, since a peer's one-hop neighbor peers vary with time in P2P MANETs, this strategy is costly in constructing and maintaining the P2P overlay. Also, using a resource's popularity to determine the number of walkers and their TTL without taking the time factor into account would have limited effect on improving the successful search rate due to the peers' strong mobility in P2P MANETs.

Jayapal et al. [25] proposed an Enhanced Service Discovery Protocol (ESDP) for MANET, where a service discovery process is composed of the cache placement, the cache consistency, the cache discovery and the cache replacement. Any peer that wants to provide a service should first advertise its service information to both the zone coordinator and the area coordinator. A zone coordinator is selected for caching the service information shared by the peers within a geographic area, and the collective information of all the zones is maintained by a single area coordinator. A minimum spanning tree is employed to manage the cache path for easing the service discovery. However, the maintenance of the zone coordinators and the area coordinator would incur much network overhead under the situation that peers are with strong mobility in MANETs. Meanwhile, using the minimum spanning tree algorithm to manage the service caches would make the problem worse.

The *k*- random walk algorithm [26, 27] attempts to reduce the search delay by having the request peer forward *k* walkers with the same query message to *k* randomly selected neighbor peers. Each walker goes along its own path in the network. When an intermediate peer receives a random walker, it checks whether the resource is available or not. If the resource is found, it returns the resource to the request peer along the path the walker came; otherwise, it forwards the walker to a randomly selected neighbor peer until the value of TTL (time to live) reaches 0.

All the above-mentioned approaches did not take the time factor into account to cluster peers or replicate resources. As mentioned in Sect. 1, in a P2P MANET, even though a peer is in the same location, its transacting *partners* may be largely different in different time intervals. Therefore, such approaches' effectiveness and efficiency cannot be guaranteed. Considering the importance of location factor for clustering peers to tackle the limitations of both the peers' short transmission range and strong mobility, this paper takes both the location and the time factors into consideration to design our resource search scheme, expecting to both improve the resource search success rate and reduce the message complexity in P2P MANETs.

3 Definitions and overview

In this section, we present several definitions and describe the transacting information management approach needed to introduce our resource searching scheme in P2P MANETs. Meanwhile, we give an overview of the components that are involved in our approach.

3.1 Related definitions and assumption

To raise the understanding of the readers and also to ease the descriptions in the following sections, we present several definitions and an assumption as follows.

Definition 1 (*Resource*) Resource refers to any service or service information, such as a file, a message or a piece of transaction information. Any node can be the holder of a resource.

Definition 2 (*Requester*) Requester is a peer that sends a request for a resource in a transaction.

Definition 3 (*Provider*) Provider is a peer that provides a resource (e.g., file) for the requester in a transaction.

Definition 4 (*Router*) Router is an intermediate peer within a path through which a request or a response can be passed between the requester and the provider.

Definition 5 (*Partner*) *Partner* is a peer to which the requester can make one-hop connection within the wireless transmission range. In other words, peer *i*'s *partners* are such peers whose locations are in peer *i*'s wireless transmission range. We can also call them the peer *i*'s one-hop neighbor peers.

Assumption 1 Any peer in the P2P MANET can obtain the information related to its current location and current time. The two metrics would be used to cluster peers and to select *partners* in the P2P MANET environments.

In a P2P MANET environment, each peer can play any of the roles of requester, provider and router according to the above definitions. We know that to improve resource lookup efficiency, how to improve the probability that the requester' request can reach the provider's location through a path of routers is a key issue to be solved in the dynamic P2P MANET environments.

3.2 Information management of a peer's *partners*

In order to cluster peers based on the locations, we should first manage the information related to a peer's *partners*. To do this, each peer locally manages

Table 1 Information management of a peer's *partners*

Location	Time	Partner	Partner's location
CL_1	T_1	P_1	L_1
CL_2	T_2	P_2	L_2
…	…	…	…

its *partners*' information with the data structure shown in Table 1, where *Location* and *Time*, respectively, represent the peer's location with the form of (x, y) coordinates and the time at which the transaction takes place; *Partner* stands for the information of *partner* with which the peer established a successful transaction. Here, a successful transaction means the peer (i.e., requester) successfully received its requested resource, including a service, a file or a piece of information; *Partner's location* represents the *partner*'s location, which is obtained with the receiving of the requested resource from the *partner*.

Note that we only record one-hop *partners*' information for a peer in Table 1, instead of the paths' information (i.e., the routers' information) and the providers' information. This is because in a dynamic P2P MANET environment, it is difficult to maintain the connectivity of the paths successfully used before due to the peers' strong mobility [8]. Since each peer has its own *partners*' information managed with Table 1, it is still easy to establish a respective path between two peers in the network. Therefore, such *partners*' information management mechanism would contribute to both coping with the peers' mobility feature and clustering peers based on locations. We will detail the location-based peers' clustering algorithm using the information listed in Table 1 in Sect. 4.

To save storage space as well as to eliminate obsolete transacting records, we use FIFO (first in first out) algorithm to maintain Table 1.

3.3 Overview of the components involved in our scheme

A P2P MANET is a highly dynamic network. Each peer in a P2P MANET has strong mobility. In such a network, the key to improve the resource search efficiency is how to efficiently and effectively cope with or adapt to the peers' movement feature. Existing researches mainly use the location-based clusters or replications to handle the peers' mobility issue. However, such approaches suffer from the problems of higher maintenance cost and higher network overhead. This is because such approaches need to maintain the cluster heads or a dominating set consisting of the peers with low mobility as mentioned in Sect. 2. Our approach tackles peers' mobility by both creating the location-based clusters and selecting the time-aware *partners*. First, the created clusters no longer need cluster heads, and thus reducing the maintenance cost. Second, the selected time-aware *partners* can guarantee that the possibility that the peers to which the requesters send requests are online in the current time interval is high, and thus reducing the propagated messages and improving the successful search rate.

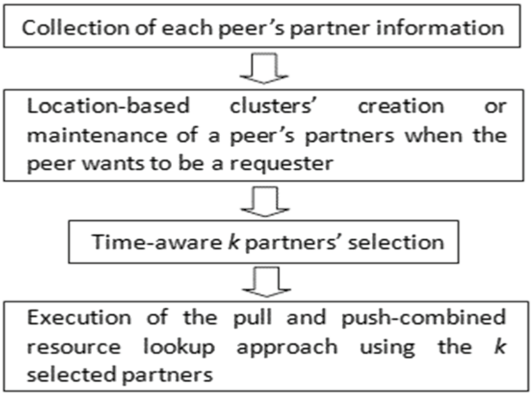

Collection of each peer's partner information
Location-based clusters' creation or maintenance of a peer's partners when the peer wants to be a requester
Time-aware k partners' selection
Execution of the pull and push-combined resource lookup approach using the k selected partners

Fig. 1 Overview of the components involved in a resource search process

Figure 1 shows the components involved in a resource search process under the *LoTiSearch* scheme. As shown in Fig. 1, our approach starts from the collection of each peer's *partner* information. If a peer accumulated enough *partners*' records (more precisely, the peer has established at least one successful transaction with its neighbors in each location determined by the peer's movement patterns), then the peer can use our approach to complete a resource search process, including the location-based clusters' creation or maintenance, the time-aware *k partners*' selection and the pull and push-combined resource lookup; otherwise, the flooding method should be used to both complete a resource search process and collect *partners*' information. Details can be found in the following.

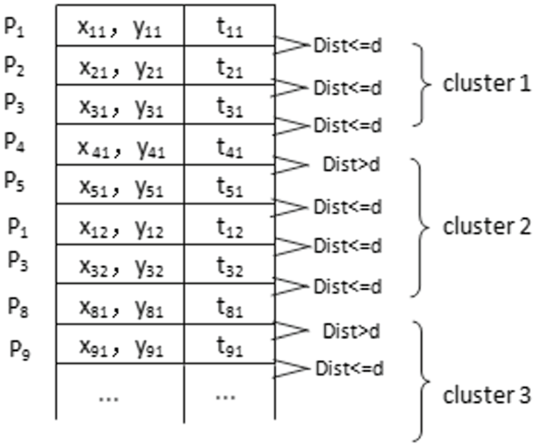
4 Location-based clusters' creation and maintenance

As mentioned above, in this paper, we improve resource search success rate by using the location and time-aware approach, including the location-based clusters' creation and the time-aware *partners*' selection. In this section, we mainly detail the location-based peers' clustering scheme. In the next section, we will describe the time-aware *partners*' selection approach.

4.1 Location-based clusters' creation

A lot of existing researches have been made on clustering peers in P2P MANETs based on locations [1, 3–5]. Here, we propose a novel algorithm suitable for clustering peers based on locations in P2P MANETs. To cluster peers based on locations, we first supply the distance calculation formula for two locations. Assume L_1 and L_2 are the two locations. Then, the distance between L_1 and L_2 is calculated with Formula (1).

$$Dist(L_1, L_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$



P_1	x_{11}, y_{11}	t_{11}
P_2	x_{21}, y_{21}	t_{21}
P_3	x_{31}, y_{31}	t_{31}
P_4	x_{41}, y_{41}	t_{41}
P_5	x_{51}, y_{51}	t_{51}
P_1	x_{12}, y_{12}	t_{12}
P_3	x_{32}, y_{32}	t_{32}
P_8	x_{81}, y_{81}	t_{81}
P_9	x_{91}, y_{91}	t_{91}
	…	…

Fig. 2 Illustration of the location-based peers' clustering

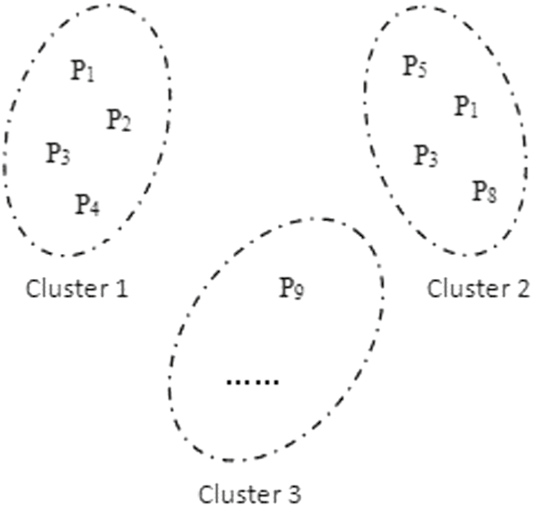


Fig. 3 Peers' distribution of Fig. 2 in space and in clusters

where (x_1, y_1) is location L_1's coordinates and (x_2, y_2) is location L_2's coordinates. Here, the distance between two points is in a plane coordinate system.

Assume the value of each peer's wireless transmission range is d. Then, Algorithm 1 shows the process of the location-based peers' clustering in P2P MANETs.

Algorithm 1 The process of creating clusters based on locations

Step 1: Assume there are n records in the list of peer m's *partners* (see Table 1), and these *partners*' locations are, respectively, denoted as L_1, L_2, …, L_n. We arrange these records in the ascending order of x^2+y^2, where x and y represent the coordinates of each *partner*'s location, L_i. The ordered records are stored in *List1*.

Step 2: From the head of *List1*, we make pairwise calculations on the distance of locations of the two peers (i.e., *partners*), say peer i and peer $(i+1)$, in the adjacent records using Formula (1). If their distance is not greater than d, then they are both put into the same cluster. Only when the distance of the adjacent peer i and peer $(i+1)$ is greater than d, do we need to create a new cluster for peer $(i+1)$ and the subsequent peers. Figure 2 illustrates the location-based peers clustering process, and Fig. 3 shows the peers' distribution of Fig. 2 in space as well as in clusters.

As shown in Fig. 2, since the distance of P_4 and P_5 is greater than d, P_4 is put into cluster 1 and P_5 is put into cluster 2. Similarly, P_8 is put into cluster 2 and P_9 is put into cluster 3. Note that there could be multiple records for a single peer even in the same cluster. Also, a peer's records could be in different clusters due to its different locations, such as P_1 and P_3 shown in Fig. 2.

Step 3: For each newly created cluster, C_j, we calculate its center's location with Formula (2).

$$Z\left(C_j\right) = \left(x = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i, y = \frac{1}{n_j}\sum_{i=1}^{n_j} y_i\right) \tag{2}$$

where n_j is the number of records in cluster C_j, x_i and y_i are the coordinates of ith record's location in cluster C_j. Here, that we calculate the center's location of each cluster is for the purpose of maintaining and adjusting the created clusters when there are newly accumulated transacting records.

The information of created clusters for each peer is managed locally with the form of Table 2, which will be used in the following sections to design our resource search approach.

In Table 2, *Cluster ID* represents the created cluster's ID and *Location of cluster's center* stands for the location of the created cluster's center. *Location*, *Time*, *Partner* and *Partner's location* are the same as mentioned in Table 1.

Existing researches indicated that each peer in a P2P network has its own interests [28–30]. This means that the possibility that the requests sent by a peer are for the same or similar resources is high. Recall that Table 1 saves the information of

Table 2 Information management of the created clusters

Cluster ID	Location of cluster's center	Location	Time	Partner	Partner's location
C_1	$Z(C_1)$	CL_{11}	T_{11}	P_{11}	L_{11}
		CL_{12}	T_{12}	P_{12}	L_{12}
		…	…	…	…
C_2	$Z(C_2)$	CL_{21}	T_{21}	P_{21}	L_{21}
		CL_{22}	T_{22}	P_{22}	L_{22}
		…	…	…	…
…	…	…	…	…	…

partners with which the current peer performed successful transactions. Under this situation and also considering the fact that each mobile device has limited transmission range, it is not difficult to imagine that the created clusters can be used to improve resource search success rate.

4.2 Incremental maintenance of the created clusters

With the increase in a peer's transactions, the created clusters for the peer should be maintained and adjusted accordingly, so as to guarantee their availability in resource searching process. To reduce computational complexity, we adopt an incremental maintenance approach to complete the clusters' adjustment.

Assume there are n' newly increased records of transactions (see Table 1) established by requester i. Then, the incremental adjustment on the created clusters for requester i is shown in Algorithm 2.

Algorithm 2 Incremental adjustment on the created clusters

Step 1: For each newly increased transaction record, we get its location information L_i and calculate the distance from L_i to each existing cluster's center, and put the record (i.e., the peer) to the cluster, say C_j, whose center is the closest to L_i. Also, we put the record to C_j' for adjusting the center's location of cluster C_j later.
Step 2: For each C_j', we calculate its center's location using Formula (3), which is similar to Formula (1).

$$Z\left(C_j'\right) = \left(x' = \frac{1}{n_j'} \sum_{i=1}^{n_j'} x_i, y' = \frac{1}{n_j'} \sum_{i=1}^{n_j'} y_i\right) \tag{3}$$

where n_j' is the number of *partners* (*partners*' records) put into C_j', and x_i and y_i are the coordinates of i^{th} *partner*'s location in C_j'.
Step 3: For each newly adjusted cluster C_j, we adjust its center's location using Formula (4).

$$Z\left(C_j\right) = \left(x = \frac{(x_i \times n_j + x_i' \times n_j')}{n_j + n_j'}, \quad y = \frac{\left(y_i \times n_j + y_i' \times n_j'\right)}{n_j + n_j'}\right) \tag{4}$$

where $Z(C_j)$ is the newly updated center's location of cluster C_j, and the other parameters are the same as mentioned in Formulae (2) and (3).
Step 4: Update Table 2 with the newly adjusted center's location for each cluster.

As mentioned above, peer movements in P2P MANETs are often repeated on a day-to-day basis [9–11]. This means we can start creating clusters for a peer after the peer joined the P2P MANET for several days (more precisely, we can start clustering peers based on locations for a peer when the peer walked through all the locations (positions) determined by its movement patterns and also it has established at least one successful transaction in each location), and the maintenance and adjustment of the created clusters can be triggered whenever the peer wants to send a request for a resource.

5 Time-aware *partner* selection approach

In Sect. 4, we depicted the location-based peers' clustering process. By using Algorithms 1–2, any peer *i*'s *partners* can be divided into clusters. These clusters are with different ranges of locations. Though we can use these clusters to improve our resource search success rate, its effectiveness cannot be guaranteed. As mentioned in Sect. 1, even though a peer is in the same location (position), it may have different *partners* in different time intervals due to the *partners*' movement patterns and the limitation of wireless transmission range. For example, in a residential quarter, the manager may have different *partners* in different time intervals, since the residents and the keepers of the residential quarter may have different working time. Therefore, we should also take the time factor into account to design resource search scheme in P2P MANETs.

The location-based clustering process is performed when a peer joined the P2P MANET for several days according to the finding that peer movements are often repeated on a day-to-day basis [9–11]. Different from this, the time factor needs to be considered only when a requester wants to send a request for a service to its *partners*. At this time, we should consider to which peers (i.e., *partners*) the request should be sent by the requester, since the request should be sent to the *partners* which should be the requester's one-hop neighbors at the current location and the current time (within the requester's wireless transmission range). In P2P networks, the random walk approach is often applied in resource search field [31], due to the fact that it can largely reduce the traffic overhead and achieve acceptable success rate in comparison with flooding method. In this paper, we also adopt the random walk algorithm to send requests and search resources. We set the number of walkers to *k*, meaning that a request should be sent to *k* one-hop neighbor peers (i.e., *partners*). Here, we discuss how to select *k partners* to which a request should be sent in Algorithm 3.

Algorithm 3 Time-aware *k partners* selection approach

Step 1: Assume requester *i* wants to send a request for a service to its *k partners*. It first determines the cluster the *partners* should be in based on its current location, L_c, with the help of Table 2. Here, we assume C_j is determined to be the cluster the *partners* should be in.
Step 2: Assume the current time is t_c. We calculate how many *partners* there are in the time interval of $[t_c - \delta, t_c + \delta]$ in cluster C_j, where δ is set to 120 s according to the finding in [10]. This can be completed with the following process.

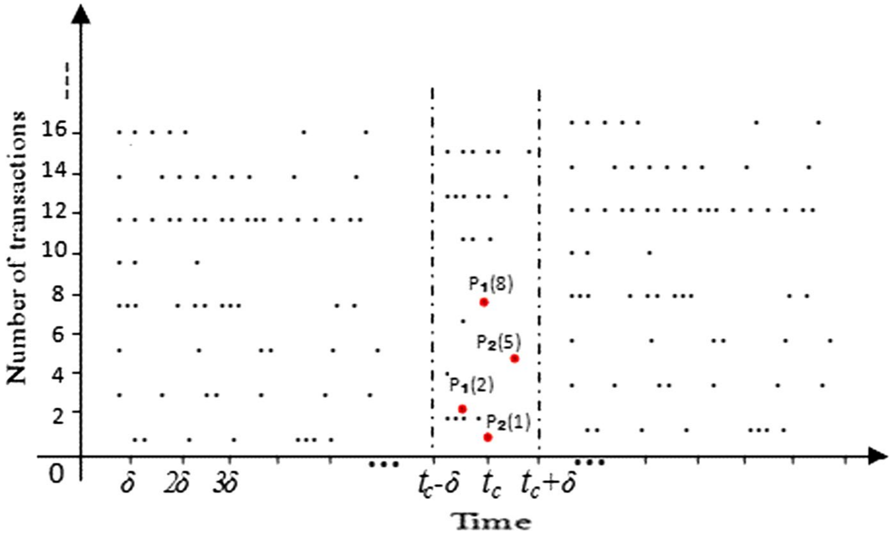


Fig. 4 The map of each *partner*'s transactions with requester *i*

Step 2.1: As shown in Fig. 4, we map the number of transactions each *partner* established with requester *i* in each time interval to a two-dimensional space. In Fig. 4, the *x*-axis represents the time of a day, which takes 120 s as a unit. The *y*-axis plots the number of transactions each *partner* established with requester *i*. From Fig. 4, we see that in the time interval of $[t_c-\delta, t_c+\delta]$, requester *i* established 8 transactions around time t_c and 2 transactions around time $t_c-\delta/2$ with *partner P1*; 1 transaction around time t_c and 5 transactions around time $t_c+\delta/2$ with *partner P2*, and so on.

Step 2.2: From Fig. 4, we know which *partners* established transactions and how many transactions they established with requester *i* in the time interval of $[t_c-\delta, t_c+\delta]$.

Step 3: With Step 2, we can obtain the number, say *p*, of *partners* with which requester *i* established transactions in the time interval of $[t_c-\delta, t_c+\delta]$. If *p* is greater than *k*, then we select the top *k partners* which performed the highest number of transactions with requester *i* in the time interval of $[t_c-\delta, t_c+\delta]$; if $0<p\leq k$, then we set $k=p$; else we set $k=0$, meaning that we should use the flooding approach to send request.

6 Pull and push-combined resource search process

In the above sections, we described how to cluster peers based on locations and how to select *k partners* for a requester with a time-aware approach based on the created clusters. The reason for doing such work in advance is because we need to deal with both the peers' mobility feature and the limitation of short radio transmission range when we want to improve resource lookup success rate in P2P MANETs.

In the following, we present the pull and push-combined resource lookup approaches, where we will discuss when and how Algorithms 1–3 are used in them.

6.1 Resource lookup with the pull approach

The pull approach is commonly applied to the resource search area in P2P MANETs. Based on the above descriptions, our resource lookup scheme with the pull approach is given in Algorithm 4.

Algorithm 4 The resource lookup scheme with the pull approach

Step 1: Assume requester *i* wants to receive a resource (e.g., a file). It first uses Algorithm 1 to perform the location-based clustering operation if the clustering operation has not been done and requester *i* has joined the network for several days, or uses Algorithm 2 to adjust and maintain the created clusters with the newly increased transaction records. Otherwise, the flooding approach is used.
Step 2: Requester *i* sends its request for a resource to its *k partners* selected using Algorithm 3 or floods its request if there are no selected *partners*.
Step 3: If a *router* (*partner*) that received the request holds the resource, it returns the resource with its location information to the requester through the routers (i.e., the path) the request came; otherwise, the *router* forwards the request to a *partner* of the *router* selected using the same approach as mentioned in Algorithm 3 if TTL (time to live) is greater than zero. Note that since each request (i.e., request message) consists of the ID of requested resource, the path (i.e., routers) through which the request is forwarded and the time at which the request is issued, the resource found in a peer can be returned to the requester through the path the request came.
Step 4: When a *partner* which forwarded the request has not received the requested resource yet after a given time elapsed (this means the resource lookup failed in this path) and also it is willing to help get the resource for requester *i* in other time intervals, the *partner* can send back a help message to requester *i* declaring that it can search the resource for requester *i* in other time intervals.
Step 5: If requester *i* received its requested resource, then it saves the transacting record to its *partners*' list as shown in Table 1; otherwise, if requester *i* received the help messages (there can be multiple help messages received by requester *i*), then it runs "resource lookup with the push approach," as detailed in Sect. 6.2.

A help message plays an important role in improving resource search success rate in P2P MANETs. As we know, a *partner* can only establish transactions with its limited *partners* in a specific time interval due to the inherent peers' mobility feature and the short wireless transmission range in P2P MANETs. This means even though a *partner* cannot find the resource for the requester in the current time interval, it may also be possible to find the resource in other time intervals, because it may have different *partners* in different time intervals. In addition, since the k selected *partners* performed the most transactions with the requester in the current time interval, their interest is more similar to that of the requester. More precisely, the requester's interest is similar to that of the peers in the paths, respectively, started from the k selected *partners* due to the fact that each peer in a P2P network has its own interests [28–30]. This means even in other location and other time interval, the possibility that the *partners*' interest is similar to that of the requester is also high, which can improve the requester's successful search rate with the help of the selected helpers. This gives us a hint to design a resource lookup scheme with the push approach, as mentioned in the following section.

6.2 Resource lookup with the push approach

As mentioned in Algorithm 4, the resource lookup with the push approach is executed when the requester failed to receive its requested resource but received the help messages (whose sender is called the *helper*) from its *partners*. To implement the push approach, each peer (i.e., each requester) needs to record the information of the total pushed transactions and the successful pushed transactions for each *partner*. Algorithm 5 lists the steps that the resource lookup with the push approach should do.

Algorithm 5 Resource lookup scheme with the push approach

Step 1: Requester i calculates each *helper*'s successful push rate using Formula (5).

$$Push\_rate_j = \frac{Succ\_push_j}{Total\_push_j} \tag{5}$$

where $Succ\_push_j$ represents the number of successful push operations completed by *helper j*, and $Total\_push_j$ is the number of all the push operations completed by *helper j*.

Note that even though a peer (i.e., a *helper*) is willing to search a resource for the requester in other time intervals, it may fail again. This is partly because the helper's *partners* do not hold the requested resource regardless of any time interval, and partly because a P2P MANET is with high churn rate, making the resource that was found by the *helper* unable to arrive at the requester.

Step 2: Requester i arranges the helpers in the descending order of their successful push rates calculated in *Step 1*, and selects the top m helpers to be the peers to search resource for requester i. Here, m is the minimum value satisfying the condition given in Formula (6).

$$\sum_{j=1}^{m} Push\_rate_j \geq 1 \tag{6}$$

Note that we should select as few helpers as possible to reduce network traffic, as long as the successful push rate can be guaranteed. To this end, we use Formula (6) to select helpers.

Step 3: Requester i informs the selected helpers that it is interested in finding the resource in a later time period.

Step 4: If a selected *helper* finds out the requested resource in other time interval, it would push the resource to the requester using a routing protocol (e.g., AODV) when the *helper* enters the requester's cluster again.

As shown in Algorithm 5, the resource lookup scheme with the push approach can make the best use of peers' mobility in P2P MANETs. Combined with the pull approach, our resource lookup scheme can achieve better resource search success rate.

7 Complexity analysis

Our approach's complexity involves the computational complexity and the message complexity. We calculate our approach's complexity based on the procedure given in Algorithm 4, since Algorithm 4 is used to complete a resource search process, where Algorithms 1–3 are also applied.

In our approach, a requester needs to create or maintain location-based clusters (using Algorithm 1 or 2) and make time-aware k *partners*' selection (using Algorithm 3). Therefore, the computational complexity of our approach is $O(p+k*q)$, where p is the average number of historical *partners* the requester holds, q is the average number of peers a cluster has, and k is the number of *partners* selected using the time-aware method.

As for the message complexity, in a resource search process, our approach needs sending at most $k*\mathrm{TTL}$ messages (in the worst case that the request passed through TTL hops for finding the resource), where k is the number of *partners* selected using the time-aware method and TTL is the maximum hops a request passes through. Hence, the maximum message complexity of our approach is $O(k*\mathrm{TTL})$, where k and TTL are the same as mentioned above.

8 Simulations and results

In this section, we evaluate the performance of our strategy, *LoTiSearch*, by simulations. Our simulations are coded in C++ as did in work [32]. In our simulations, AODV [33] is selected as the MANET routing protocol, though our strategy is independent of the MANET routing protocol. The simulation area is set to 600 × 800 m^2, which is divided into 48 sub-areas. Each sub-area is 100 × 100 m^2. We, respectively, employ 300 peers and 600 peers to perform the simulations. Two wireless transmission ranges are used, respectively, to perform the simulations, one is set to 20 m and the other is set to 50 m. Each peer has 3 interests and moves regularly among 6 sub-areas. Each simulation runs 100 cycles and 1000 cycles, respectively. In a simulation cycle, we randomly select 10 peers to issue resource requests in every cluster, and the resource requested by a peer is related to the peer's interests. As mentioned above, such simulation settings are based on the findings that peer movements are often repeated on a regular basis and each peer has its own interests in P2P MANETs [9–11]. Considering such realistic situation, we allow that each peer can move randomly with the probability of 10% without following the regular movement. The maximum hop count (i.e., TTL) of forwarding a request is set to 5.

We assume a file sharing system as the application scenario of our strategy and each peer in the network can play the roles of a requester and a provider. All the files are classified into 20 kinds of interests and randomly distributed on the peers. The above-mentioned simulation settings are listed in Table 3.

We examine our strategy's performance focusing on the successful search rate and the propagated messages in comparison with the work of *2P-Lookup* [24], ESDP [25] and the flooding approach [23]. These approaches have already been detailed in Sect. 2. Since the standard flooding approach consumes too much network traffic, we randomly choose 60% one-hop neighbor peers (i.e., *partners*) to flood the requests according to the work in [23]. To simulate the work of *LoTiSearch*, ESDP and *2P-Lookup*, we first run 20 cycles of the simulations to collect basic interaction data or perform cache placement. Then, for the *LoTiSearch* scheme, Algorithm 1 is executed to conduct the location-based peers clustering operation, and from then on our strategy is used to adjust and maintain the created clusters using Algorithm 2 and search files using Algorithms 3–5. For the *2P-Lookup* approach, the resource popularity is calculated to determine the number of walkers and the value of TTL. Also, for the ESDP approach, the minimum spanning tree is established to manage the service caches maintained by different zone coordinators. Data collection is performed at the end of each simulation cycle. Each simulation runs ten times, and the average value is reported as the simulation result.

8.1 Successful search rate

In this simulation, we focus on our strategy's successful search rate, which is defined as the ratio of the number of successful search requests to the number of total search

Table 3 Simulation parameters

MANET routing protocol	AODV
Simulation area size	600 m × 800 m
Number of sub-areas	48
sub-area size	100 m × 100 m
Number of peers	300, 600
Movement patterns	6 sub-areas
Wireless transmission range	20 m, 50 m
Each peer's interests	3
Total interests	20
TTL	5

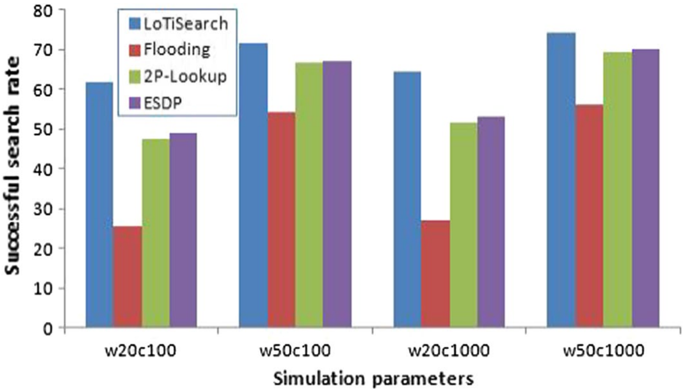


Fig. 5 Successful search rates under 300 peers

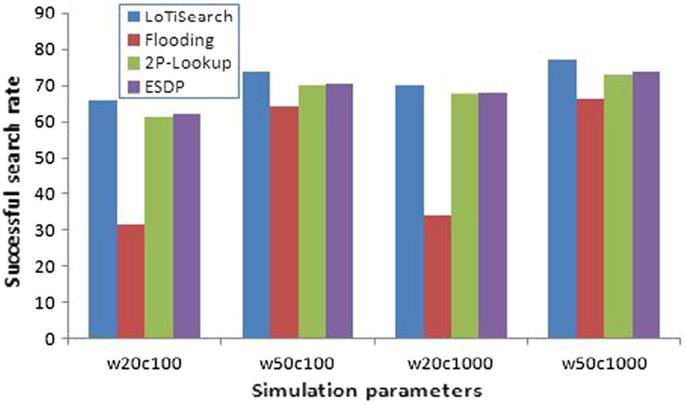


Fig. 6 Successful search rates under 600 peers

requests. A higher successful search rate implies that the requesters can get more satisfied search results.

In Figs. 5 and 6, "w20c100" represents the simulation parameters used to get the corresponding simulation result, where w20 means the peer's wireless transmission range is set to 20 m, and c100 implies that the corresponding simulation runs 100 cycles.

As shown in Figs. 5 and 6, a shorter transmission range results in a lower successful search rate regardless of which strategy is used. This is because a shorter transmission range makes a peer own fewer one-hop neighbor peers, lowering the possibility that the peer can successfully search for its requested resources. Among the four strategies, the *LoTiSearch* strategy always maintains a higher successful search rate. After the location-based clusters have been created, *LoTiSearch* uses the time-aware approach to select *k partners*, which can guarantee with a high probability that the selected *partners* are both being online and within the peer's wireless transmission range in the current time interval. Furthermore, since we adopt the random walk algorithm that uses the selected *k partners* to send requests to implement our pull and push-combined resource lookup algorithm, our strategy's successful search rate can be improved. The *2P-Lookup* approach [24] wants to improve the successful search rate by both constructing the overlay network based on proximity and determining the number of walkers and their TTL using resource popularity. However, since a peer in a P2P MANET has strong mobility, its neighbor peers would vary with time. In such situation, the resource popularity computed without taking the time factor into account would limit the approach's effectiveness in improving its successful search rate. The successful search rate of the ESDP approach is slightly higher than that of *2P-Lookup* for the reason that it caches the service information in different zone coordinators and employs the minimum spanning tree to manage the service caches located in different geographic areas. Note that the higher successful search rate of the ESDP approach comes from its service cache management which would incur much maintenance overhead. As for the flooding approach, since the 60% of all neighbor peers are selected randomly without using the location-based clusters and the time-aware *partner* selection approach, this lowers its successful search rate.

Figures 7 and 8 show the increased successful search rate by using the helpers. The increased successful search rate is the ratio of the successful search rate achieved by using the helpers to the overall successful search rate. Note that the increased successful search rate shown in Figs. 7 and 8 is achieved only using one helper to resend the request when it moves to its next cluster determined by its movement patterns. Combined with the data shown in Figs. 5 and 6, we see that the lower the overall successful search rate is, the higher the increased successful search rate achieved by using helpers is, and vise versa. Therefore, using helpers in highly dynamic P2P MANETs can improve resource search success rate.

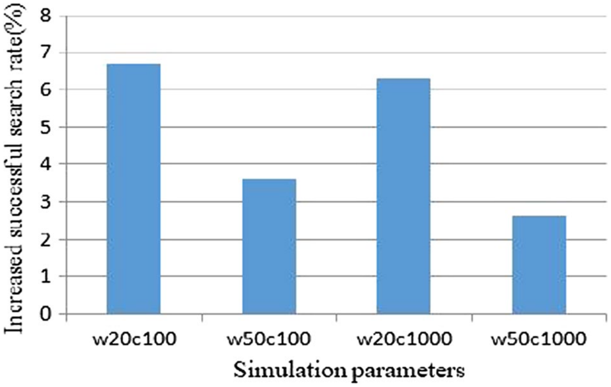


Fig. 7 The successful search rate increased by using the helpers under 300 peers

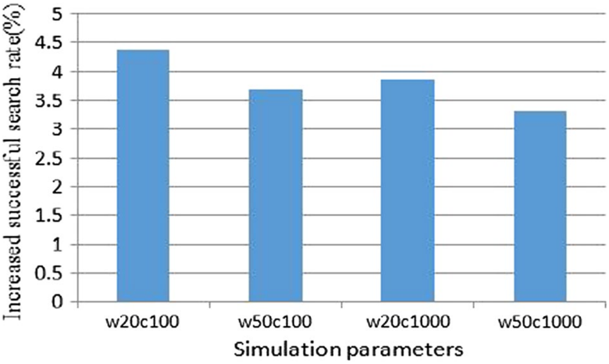


Fig. 8 The successful search rate increased by using the helpers under 600 peers

8.2 Propagated messages

This simulation examines the propagated messages of our resource search strategy in comparison with the work of *2P-Lookup*, ESDP and the flooding approach. This metric can be used to evaluate the message complexity of the resource search strategies, since the higher the number of propagated messages is, the higher the network overhead is.

As shown in Figs. 9 and 10, the *LoTiSearch* strategy always propagates fewer messages than other three strategies. In our strategy, we adopt the random walk algorithm to send a request to *k* one-hop neighbor peers (i.e., *partners*), which are selected by using Algorithm 3. From the definition of Algorithm 3, we see that the *k* selected neighbor peers are those which have established the most transactions with the requester that issued the request in the current time interval. Therefore, we can say that the *k* selected neighbor peers are those whose online probability is the highest among all the *partners* in the current time interval, thus making the number of

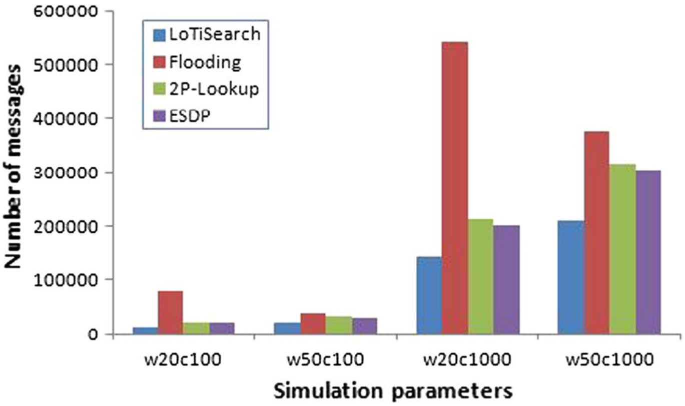


Fig. 9 Propagated messages under 300 peers

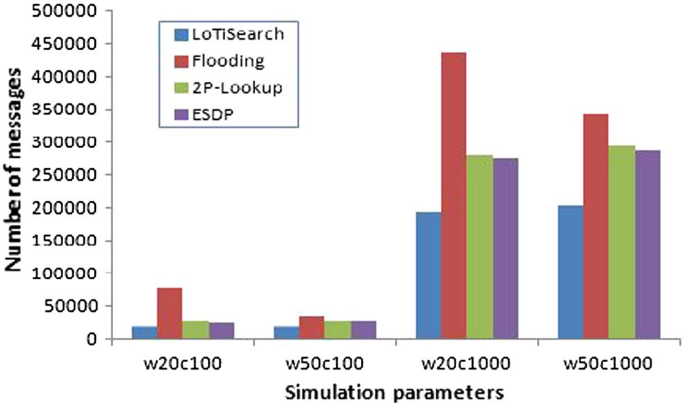


Fig. 10 Propagated messages under 600 peers

propagated messages lower. Note that if a request is sent or forwarded to the *partners* with lower online probability, not only the successful search rate would be lower, but also the propagated messages would be largely increased, due to the need of resending the request. The *2P-Lookup* approach uses the resource popularity to determine the number of walkers and their TTL for using the random walk approach under the assumption that the higher the popularity of a resource is, the higher the probability of finding it in a close area is. However, since *2P-Lookup* calculates the resource popularity without taking the time factor into account, the number of its propagated messages is higher under the finding that each peer has its own movement patterns in P2P MANETs. From Figs. 9 and 10, we see that the number of the propagated messages of the ESDP approach is slightly lower than that of the *2P-Lookup* approach. However, the maintenance overhead of ESDP for managing the zone coordinators and area coordinator as well as the minimum spanning tree is much higher than that of *2P-Lookup* in a highly dynamic P2P MANET, which is not included in the calculation of the propagated messages. As for the flooding approach, the number of its propagated messages is the highest among the four strategies, though we flood the requests to only 60% neighbor peers. The blindness of selecting 60% neighbor peers to flood the requests makes it own lower successful search rate and higher propagated messages.

9 Conclusions

In this paper, we discussed how to improve resource search performance in mobile P2P ad hoc networks. Considering the fact that each peer is with strong mobility and limited wireless transmission range in P2P MANET, we first proposed a location-based peers' clustering mechanism and a time-aware *partners*' selection approach, and then, we have presented a combined resource search scheme employing both the pull and push approaches. In the pull approach, a requester's service request is sent and forwarded to *k* one-hop neighbor peers, which are selected based on the number of their past transactions established with the requester in the current time interval, thus making the best use of the finding that peer movements are often repeated on a day-to-day basis. In the push approach, a *helper* can find the resource for its requester in other time intervals, thus making the successful search rate increased. The simulation results indicated that our resource search scheme could achieve higher successful search rate and lower traffic overhead. In the future work, we will focus our efforts on improving the resource search performance under the presence of malicious peers by using the trust model in P2P MANETs.

References

1. Arunachalam Ajay, Sornil Ohm (2016) Reducing routing overhead in random walk protocol under MP2P network. Int J Electr Comput Eng 6(6):3121–3130
2. Imaizumi N, Kobayashi K, Utsu K et al (2016) A study on effective flooding over MANET based on exchange of neighbor information. J Supercomput 72:1237–1245
3. Keke Gai, Meikang Qiu (2018) Reinforcement learning-based content-centric services in mobile sensing. IEEE Netw 2(4):34–39
4. Al Mojamed M, Kolberg M (2016) Structured Peer-to-Peer overlay deployment on MANET: a survey. Comput Netw 96:29–47
5. Atsan Emre, Özkasap Öznur (2013) SCALAR: scalable data lookup and replication protocol for mobile ad hoc networks. Comput Netw 57:3654–3672
6. Sucasas Victor et al (2016) A survey on clustering techniques for cooperative wireless networks. Ad Hoc Netw 47:53–81
7. Chatterjee M, Das SK, Turgut D (2002) WCA: a weighted clustering algorithm for mobile ad hoc networks. Cluster Comput (Special Issue on Mobile Ad hoc Networks) 5:193–204
8. Ludvik B, Niederle B, Roka R, et al (2005) Searching for resources in MANETs—a cluster based flooding approach. In: ICETE 2005-Proceedings of the Second International Conference on E-Business and Telecommunication Networks, Reading, UK, pp 236–245
9. Henderson T, Kotz D, Abyzov I (2008) The changing usage of a mature campus-wide wireless network. Comput Netw 52(14):2690–2712
10. Orlinski M, Filer N (2013) The rise and fall of spatio-temporal clusters in mobile ad hoc networks. Ad Hoc Netw 11:1641–1654
11. Ma W, Fang Y, Lin P (2007) Mobility management strategy based on user mobility patterns in wireless networks. IEEE Trans Vehic Technol 56(1):322–330. https://doi.org/10.1109/TVT.2006.883743
12. Yunoki S, Tamaki S, Takada M, Iwaki T (2013) Remote energy monitoring system by cellular network. Electron Commun Jpn 96(11):35–43
13. Rahman M, Harmantzis F (2004) IEEE 802.11 Inter-WLAN mobility control with broadband supported distribution system integrating WLAN and WAN. Telecommunications and networking–ICT 2004: lecture notes in computer science; 3124; Fortaleza, Brazil
14. Lacuesta R, Penalver L, Fernandez-Sanz L, Lloret J, Garcia M (2009) Software requirements for ubiquitous Ad hoc mobile networks: an example of a bluetooth application. In: 2009 Fourth International Conference on Software Engineering Advances (ICSEA 2009): Porto, Portugal 20–25 Sept 2009; Porto, Portugal
15. Pushpalatha M, Ramarao T, Venkataraman R (2014) Applicability of sub graph centrality to improve data accessibility among peers in MANETs. Peer-to-Peer Netw Appl 7:129–146
16. VicterPaul P, Rajaguru D, Saravanan N, Baskaran R, Dhavachelvan P (2013) Efficient service cache management in mobile P2P networks. Future Gener Comput Syst 29:1505–1521
17. Liu Y, Xiao L, Liu X et al (2005) Location awareness in unstructured Peer-to-Peer systems. IEEE Trans Parallel Distrib Syst 16(2):163–174
18. Xu Wu (2012) A distributed trust management model for mobile P2P networks. Peer-to-Peer Netw Appl 5:193–204

19. Wang Ching-Sheng, Chen Chien-Liang, Deng Der-Jiunn (2015) P2P-based mobile navigation system with location service. Peer-to-Peer Netw Appl 8:22–31
20. Mondal A, Madria SK, Kitsuregawa M (2006) CLEAR: an efficient context and location-based dynamic replication scheme for mobile-p2p networks. Proc Dexa 4080:399–408
21. Yang Kun, Todd Chris, Li Jie et al (2009) A location-based service advertisement algorithm for pervasive service discovery in wireless mobile networks. Wirel Commun Mob Comput 9:845–858
22. Kniess J, Loques O, Albuquerque CVN (2015) Service discovery with time constraints in mobile ad hoc networks. Earth Sci Inform 8:439–452
23. Arunachalam Ajay, Sornil Ohm (2015) Issues of implementing random walk and gossip based resource discovery protocols in P2P MANETs and suggestions for improvement. Proc Comput Sci 57:509–518
24. Seddiki Manel, Benchaïba Mahfoud (2016) 2P-lookup: popularity and proximity based P2P lookup mechanism over MANETs. J Netw Comput Appl 71:181–193
25. Jayapal C, Jayavel S, Sumathi VP (2018) Enhanced service discovery protocol for MANET by effective cache management. Wirel Pers Commun 103:1517–1533
26. Himali DMR, Prasad SK (2011) SPUN: a P2P Probabilistic search algorithm based on successful paths in unstructured networks. In: IEEE international symposium on parallel and distributed processing workshops, pp 1610–1617
27. Mahdi G, Mohammad R. M, Ali. MS (2013) A new version of k-random walks algorithm in Peer-to-Peer networks utilizing learning automata. In: 2013 5th International Conference on Information and Knowledge Technology, Shiraz, Iran, 5, pp 1–6
28. Rehman RA, Hieu TD, Bae HM et al (2016) Robust and efficient multipath interest forwarding for NDN-based MANETs. In: 2016 9th Ifip Wireless and Mobile Networking Conference, Colmar, France, July 11–13, pp 187–192
29. Hsiao H-C, Su H-W (2012) On optimizing overlay topologies for search in unstructured peer-to-peer networks. IEEE Trans Parallel Distrib Syst 23(5):924–935
30. Lin KC-J, Wang C-P, Chou C-F, Golubchik L (2010) SocioNet: a social-based multimedia access system for unstructured P2P networks. IEEE Trans Parallel Distrib Syst 21(7):1027–1041
31. Bisnik N, Abouzeid AA (2007) Optimizing random walk search algorithms in P2P networks. Comput Netw 51(6):1499–1514
32. Agustinus BW, David T, Wenny R et al (2013) Mobile peer-to-peer data dissemination in wireless ad-hoc networks. Inf Sci 230:3–20
33. Perkins C, Belding-Royer E, Das S (2000) Ad hoc on-demand distance vector (AODV) routing. http://www.researchgate.net/publication/2440375

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.