

Seeker-Assisted Information Search in Mobile Clouds

Suzan Bayhan
HIIT, Aalto University
Finland
bayhan@hiit.fi

Esa Hyytiä
COMNET, Aalto
University, Finland
esa@netlab.tkk.fi

Jussi Kangasharju
University of Helsinki
Finland
jakangas@cs.helsinki.fi

Jörg Ott
COMNET, Aalto
University, Finland
jo@netlab.tkk.fi

ABSTRACT

The increase in the size of *mobile cloud* as well as the volume of information necessitates efficient search mechanisms for finding the searched information or the target node. In this paper, we focus on search mechanisms to retrieve information from within a mobile cloud in which nodes have intermittent connectivity and hence operate on a *store-carry-forward* manner. We design an opportunistic search scheme in which the searching node spreads a limited number of replicas of the query to the nodes it meets and these nodes, so called *seekers*, perform the search on behalf of the searching node. We assume that nodes are grouped into communities based on their interest profiles, and seekers use this community information to forward the data and the query to the right community—the community that is more likely to store the searched content. Since people store and search for similar information in the scope of their interest, the nodes in the same community as the searching node have higher probability to store the searched content. We model this *seeker-assisted search* scheme as a continuous time Markov process and analyze its performance under various inter-community/intra-community meeting rate, number of replicas, and network population. Our analysis shows that seeker-assisted search achieves a good balance between the search response time and search cost compared to the two extremes of epidemic search and direct delivery search.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Store and forward networks*

Keywords

Mobile opportunistic networks, mobile search, interest similarity.

1. INTRODUCTION

With the proliferation of mobile devices, the Internet grows tremendously in its size as well as the heterogeneity of its components. Fixed and mobile users alike usually access a wide range of services offered by a (increasingly) cloud-based ecosystem. While

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MCC'13, August 12, 2013, Hong Kong, China.

Copyright 2013 ACM 978-1-4503-2180-8/13/08 ...\$15.00.

fixed users can largely rely on their Internet access via the well-connected always-on network infrastructure, mobile users may find the cloud to be only weakly connected and face disruptions or limited access performance. Mobile users may conspire to create a smaller *mobile cloud* among colocated devices by forming mobile opportunistic networks, also dubbed *Pocket Switched Networks (PSNs)*, among each other using short-range radio links (e.g., Bluetooth or WLAN). PSNs enable communication in spite of node mobility and resulting disconnections between nodes by using the *store-carry-forward* paradigm as known from (mobile) delay-tolerant networking (DTN), which moves messages one-hop at a time (and thus does not require instant end-to-end connectivity as the Internet does). When nodes move, they physically carry messages along, thereby enhancing connectivity. PSNs may thus form a viable alternative to facilitate communication even in the (temporary) absence of infrastructure connectivity, but their robustness comes with different constraints.

Due to the intermittent connectivity and frequent change in network topology, nodes are usually not able to obtain a network-wide coherent view of the network state at any given point in time. Solutions designed for connected networks (such as MANETs or infrastructure networks), are bound to fail. Instead, protocols and algorithms are required that work with partial and are able to tolerate outdated information. Such protocols are designed leveraging the dynamics of the network, e.g., the mobility patterns of nodes, pairwise node meetings based on the underlying social relations [17], and yield probabilistic rather than deterministic results.

We can broadly group previous work in two groups: (1) information dissemination to a certain group of nodes (or as many nodes as possible) [10, 18] and (2) routing messages from a source node to a specific destination node [1, 7, 12]. In both cases, the target(s) may be identified by a name/address, which is the common case, or a set of attributes, such as a query for a piece of content. However, as the mobile cloud gets bigger in size and the volume of information grows, efficient mechanisms for finding the respective target—node or data—become more important.

In this paper, we focus on search mechanisms to retrieve information from within such a mobile cloud. Searching for content in a PSN is nontrivial as, compared to search engines in the Internet, there is no central database to store any search index and the nodes holding the content may not be available for deterministic crawling in the first place. Therefore, instead of relying on a dedicated third party provider such as Google, nodes build their own information base to seek information at the relevant nodes. Content may be sought actively *on demand* but, due to the potentially inherently delay-tolerant nature of a PSN, we rather envision that information gathering will often be decoupled from actual user interactions and happen *in the background*. The former may happen

when a user, e.g., wants to query its immediate surroundings for video clips of a given keyword combination (inexact matches) and gather all matching results over a period of time. The latter would include web requests branched out to neighboring nodes that could have the sought objects (exact query).

We focus on an opportunistic network in which nodes move around a lot and where the nodes are sparsely distributed so that contacts are rare and isolated events. Some nodes, termed *seekers*, assist the searching node to find and reach the content it looks for. We assume that people with similar interests meet more frequently compared to two random people, and information is more likely to exist at a node with a similar interest. With this in mind, we design a search scheme exploiting community membership based on interest similarities. Then, we model this search scheme as a continuous time Markov process and analyze the time for search completion. The results show that the proposed scheme strikes a balance between the cost and the expected response time.

2. RELATED WORK

Information discovery in DTNs has not been well-investigated except a few works [3, 6, 15]. Search, being inherently a routing problem with a grand challenge of the uncertainty of the destination node, has to consider all problems related to routing in DTNs. Hui *et. al.* [6] apply the biological phenomenon of *osmosis* for spreading the file lookup query. After the query reaches the destination, the generated response routes back to the searching node via following the traces (i.e., the query) on the nodes. This scheme is mostly used as a reference scheme to set the upper bound in performance and the lower bound on the cost: the message is flooded in the network guaranteeing it to reach the destination at the expense of high burden on the network.

Pitkänen *et al.* [15] design various query forwarding policies ran by a node that has received the query and performed a local search in its buffer. Since a query must not live in the network forever, the authors also provide search termination schemes, e.g., hop-count limitation, time-to-live. Different from [15], we introduce the community concept into our model and use this information to route the queries toward the members of a specific community. Community refers to a group of nodes with noticeably higher in-group interactions (e.g., contacts) compared to the interactions with the nodes outside the group. This aspect of a network can be discovered using various techniques on the graph representation of the network, e.g., successive removal of high betweenness edges [4], analyzing number of contacts and their durations for overlapping communities [8] or density of edges in a group of nodes for dynamically evolving networks [13]. Community information is incorporated into data forwarding schemes in various previous works such as [7]. In [7], the source node and the relaying nodes first forward the data towards the highly popular nodes in the network-wide scale, i.e., have high *betweenness centrality*, till reaching a member in the same community as the destination node. Next, data is forwarded only via the members of this community and towards the popular nodes in the community-wide scale till it reaches the destination (or expires). The principal difference of our work is that since the destination node(s) holding the content is a mystery to the network, we use this community information only for controlling the spread of query towards a certain community.

Similar to our interest-based community formation, Fan *et. al.* [3] introduce a *geo-community* concept wherein nodes sharing common interests form a community and each community is associated with a geo-location, e.g. office, gym, or conference venues. However, this concept is a bit strict in the sense that a node is at a (geo-)community, only when it is at the specific area. In contrast

to the volatile community concept in [3], node’s community membership in our model is static independent of its spatiality. Besides, a searching node in [3] knows the destination of the search due to the assumed one-to-one mapping between an interest (i.e., query’s subject) and a geo-community. In our model, only assumption is that the searched content is more probable to be stored by a node in the same community with the searching node. [3] designs a knapsack based relay selection scheme utilizing the estimation on each node’s visiting the destination geo-community and probability of meeting again with the searching node.

3. SYSTEM MODEL

We consider a network in which nodes are highly mobile and have communication opportunities only when they are in contact, i.e., one node is in the communication range of the other. A node n_s initiates a search for obtaining some information defined by a *query* q . We call the searched content *response* and denote it by r . Due to the unavailability of a catalogue of information in the network, the node seeks in an opportunistic way: it asks for the content from the nodes it encounters and also may replicate the query to other nodes. Replication of the query is for speeding the search process by exploiting the mobility of diverse nodes. The nodes holding the query become informed about the search and thereby when they meet the response, they can retrieve it (or meta-information and location of the information) with the aim of delivering r to n_s . On the other hand, replicating the query without limits is undesirable as it floods the network. Hence, n_s can replicate the query to a limited number of copies [16]. Additionally, n_s can decide on which encountered node to transmit a copy of the query. The search completes when n_s meets a node that holds a valid response.

In this work, we assume that nodes have some interest profiles that can be represented by various interest categories, e.g., arts, science. Nodes with similar interests form a *community*. Suppose that the nodes are grouped into two as Community 1 denoted by C_1 and Community 2 denoted by C_2 based on their interest profiles. We do not focus on community detection, however methods provided in [8] can be tailored for distributed community detection in our model. For the tractability of analysis, similar to [5], we assume that inter-contact times are exponential i.i.d. random variables with a mean $1/\mu_i$ in a community and with a mean $1/\mu_x$ between communities. However, the research on mobility modeling in DTNs also argues that exponential distribution of pairwise inter-contact times fall short of representing human mobility. References [2, 9, 14] provide an elaborate discussion on mobility models for opportunistic mobile networks.

Suppose that members of the same community meet more frequently compared to two nodes in different communities.¹ Hence the following relation holds: $\mu_i > \mu_x$. Besides, we assume that a node searches for content in its interest profile and the related content is more likely to be stored by a node in the same community. We believe that these assumptions reflect the reality as people’s whereabouts depend on their interests, e.g., a person interested in arts may spend more time in neighborhood of the art galleries whereas a person interested in sports might spend more time in a gym. Additionally, people search for information related to their interest profiles and another node with similar interests might probably have this searched information. All the abovementioned assumptions can also be explained by *homophily phenomenon* [11] which states that people with “similar” properties have high ten-

¹In fact, communities could be vaguely defined by saying that the nodes meeting frequently belong to a same cluster.

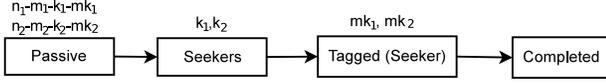


Figure 1: High level view of the search process.

dency to befriend and also have higher interaction with each other compared to less similar ones.

4. SEEKER ASSISTED SEARCH (SAS)

Let n denote the number of nodes excluding the one who executes a search operation. Of these n nodes, n_1 nodes are in C_1 and n_2 nodes are in C_2 . Apart from the searching node itself, there are four types of nodes in each community: passive nodes (P), seeker nodes (S), tagged nodes (T), tagged seekers (TS). Below, we define each node type:

Searching node (n_s): This is the node that initiates a search. Searching node denoted by n_s is assumed to be in C_1 . In order not to flood the network with too many copies of the query, the searching node generates (at most) $M+1$ copies of the query. It delivers these copies to other nodes to accelerate the search process. In particular, the source node passes one query at a time to nodes it meets until it has only one copy left for itself.

Passive node (P): A node that neither holds the query nor carries the searched content is a *passive node*.

Tagged node (T): It is a node that carries the content in its buffer matching the query initiated by n_s . When n_s encounters a tagged node, search is successfully completed.

Seeker node (S): A node that holds the query but is not the searching node is called a *seeker*. A seeker node is aware of the search but does not own the related content. A passive node turns to a seeker node if it meets with n_s , and n_s transmits a copy of the query to this node. It helps n_s to seek for information. A seeker node may pass its copy of the query to a passive node upon encounters.

Tagged seeker node (TS): A seeker node if it also holds the searched content is a *tagged seeker node*. We assume that TS does not forward the query any further.

We refer to our search scheme as *seeker-assisted search (SAS)* as the seekers assist the searching node via carrying the query, spreading it, and also retrieving the valid response from the tagged nodes for further delivery to the n_s . We use community index i to refer to each node type with its community, e.g, P_i is a passive node in C_i . Let m_i denote the number of tagged nodes, mk_i the number of tagged seekers, and k_i the number of seekers in C_i at time t , $i = 1, 2$. Consequently, the state of the network is defined by a 7-tuple,

$$\mathbf{z} = (m_1, k_1, mk_1, m_2, k_2, mk_2, c),$$

where c is the number of remaining copies held by n_s other than its own copy. As there are n_i nodes in C_i , the following conditions are always valid: $m_1 + k_1 + mk_1 \leq n_1$ and $m_2 + k_2 + mk_2 \leq n_2$. Additionally, number of copies of the query in the seekers and tagged seekers at a time cannot exceed M . So also the following always holds: $k_1 + k_2 + mk_1 + mk_2 \leq M$. Due to our assumption on the location of searched content, i.e, it is more probable to be in the same community as the searching node, C_1 in this case, seekers and tagged seekers try to move the query towards the members of C_1 . Hence while a (tagged) seeker in C_2 does not pass

Table 1: Summary of state changes triggered by meetings between each node type. P_i : passive node in c_i ; S_i : seeker in c_i ; T_i : tagged node in c_i ; TS_i : tagged seeker in c_i . Only changes in state $(m_1, k_1, mk_1, m_2, k_2, mk_2, c)$ are denoted in the table. S_{end} : search ends; ‘-’: no change in the state.

Community 1 intra-community meetings					
	n_s	P_1	S_1	T_1	TS_1
P_1	$k_1+1, c-1$	-	-	-	-
S_1	-	-	-	mk_1+1	mk_1+1
T_1	S_{end}	-	mk_1+1	-	-
TS_1	S_{end}	-	mk_1+1	-	-
Community 2 intra-community meetings					
		P_2	S_2	T_2	TS_2
P_2	-	-	-	-	-
S_2	-	-	-	mk_2+1	mk_2+1
T_2	-	-	mk_2+1	-	-
TS_2	-	-	mk_2+1	-	-
Inter-community meetings					
	n_s	P_1	S_1	T_1	TS_1
P_2	$k_2+1, c-1$	-	-	-	-
S_2	-	k_1+1, k_2-1	-	mk_2+1	mk_2+1
T_2	S_{end}	-	mk_2+1	-	-
TS_2	S_{end}	mk_1+1, mk_2-1	mk_1+1	-	-

the query to a passive node in C_2 , it pushes it to a passive node in C_1 . When the search starts, i.e., $t = 0$, the system is in state $(m_1, 0, 0, m_2, 0, 0, M)$.

Fig. 1 depicts a simplified high level view of the search operation. The passive nodes become seekers upon meeting the searching node. Next, the seekers become tagged seekers when they meet the tagged nodes and obtain a copy of a valid response. Finally, n_s meets either a tagged seeker or a tagged node, and search completes. A more detailed search operation can be found in Fig. 2.

The fundamental idea is that the seekers temporarily increase the number of nodes carrying the query and, when they meet a tagged node, they also increase the number of nodes knowing that they carry an answer to an active query. This, in turn, increases the likelihood that n_s meets either a tagged node or a tagged seeker, thus leading to a shorter expected time to acquire the response. It should be noted that even though the query is replicated from a seeker to passive nodes, the reply is provided only when n_s meets a tagged (seeker) node; the reply is not forwarded over multiple hops between nodes.

There are similarities between SAS and spray-and-wait [16] in that they both create a (small) number of replicas of the message and send it to neighboring nodes. However, the differences are that in spray-and-wait, the nodes receiving the message will only deliver it to the recipient when meeting it. In SAS, the nodes receiving the message (seekers) may pass their copy of the message to passive nodes, turning them into seekers. Furthermore, spray-and-wait is intended for unidirectional communication (unless a similar operation is performed in reverse), whereas SAS is intended for discovering content, i.e., a successful operation requires n_s to meet a tagged (seeker) node to retrieve the searched content. Our use of communities to determine when to replicate queries is another difference, however it could be considered a ‘‘spraying’’ heuristic, as discussed in [16].

We model the SAS search operation as a continuous time Markov process with the state represented by the 7-tuple \mathbf{z} . We first identify the meetings between nodes that lead to a state transition and the corresponding transition rates. Next, we derive the mean time to obtain information by calculating the mean time to meet a tagged node or a tagged seeker node. Table 1 lists all the state transitions

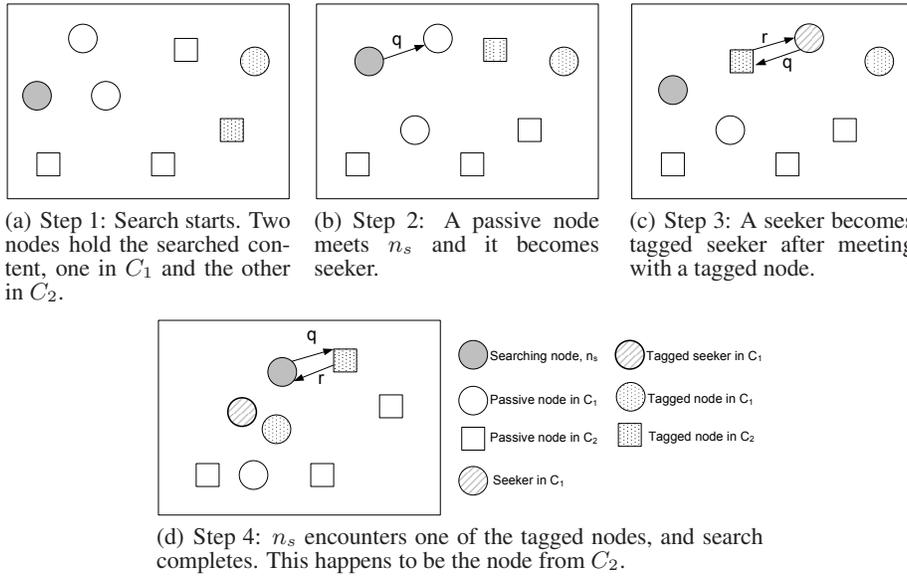


Figure 2: Summary of search steps

corresponding to meetings among different types of nodes. Please notice the difference between the state change upon P_1 meets S_2 and TS_2 compared to the case P_2 meets S_1 and TS_1 . The transition rates from state \mathbf{z} are as follows:

1. n_s encounters one of the tagged nodes $-T_1, T_2, TS_1, TS_2-$, and search ends:

$$\lambda_{\text{end}} = (m_1 + mk_1)\mu_i + (m_2 + mk_2)\mu_x.$$

2. P_1 becomes S_1 via n_s :

$$\lambda_{P_1, S_1, n_s} = (n_1 - m_1 - k_1 - mk_1)\mu_i.$$

3. P_1 becomes S_1 via S_2 :

$$\lambda_{P_1, S_1, S_2} = k_2(n_1 - m_1 - k_1 - mk_1)\mu_x.$$

4. P_1 becomes TS_1 :

$$\lambda_{P_1, TS_1} = mk_2(n_1 - m_1 - k_1 - mk_1)\mu_x.$$

5. P_2 becomes S_2 :

$$\lambda_{P_2, S_2} = (n_2 - m_2 - k_2 - mk_2)\mu_x.$$

6. S_1 becomes TS_1 :

$$\lambda_{S_1, TS_1} = k_1(m_1 + mk_1)\mu_i + k_1(m_2 + mk_2)\mu_x.$$

7. S_2 becomes TS_2 :

$$\lambda_{S_2, TS_2} = k_2(m_1 + mk_1)\mu_x + k_2(m_2 + mk_2)\mu_i.$$

Note that with SAS, the data is replicated to the seekers from the tagged nodes, and to passive nodes of C_1 from the tagged seekers of C_2 . Moreover, a seeker of C_2 gives up the query to a passive node of C_1 . With the above-listed state transitions, the mean absorption time, i.e., the mean time to complete a search, can be computed recursively. Moreover, some asymptotic cases can be easily deduced. For example, with $n_1 \gg m_1 + m_2$ the system reduces to a one-dimensional Markov process after a short initial

transient, where, e.g., the number of (C_1) seekers is the state variable. The performance analysis of this one-dimensional process is then straightforward.

We can also compute the cost of the search as the number of content replications. In the next section, we analyze the effect of intra-community and inter-community meeting rates (μ_i, μ_x), number of replicas of the query (M), and the location of the content, e.g. C_1, C_2 , or both have the searched content at the time of search initiation (e.g., m_1 and m_2).

A convenient measure of time is the mean time-interval between contacts of n_s and other nodes, $1/\mu_s$ where

$$\mu_s = n_1\mu_i + n_2\mu_x. \quad (1)$$

Consequently, the time-interval between two contacts in the whole network $1/\mu_{\text{tot}}$ is:

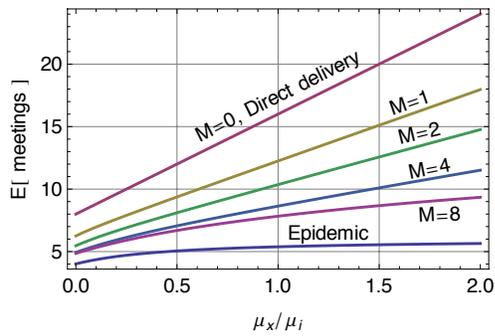
$$\mu_{\text{tot}} = \frac{n_1(n_1 - 1) + n_2(n_2 - 1)}{2} \mu_i + n_1 n_2 \mu_x + \mu_s. \quad (2)$$

where the first two terms represent the contacts between two nodes excluding n_s .

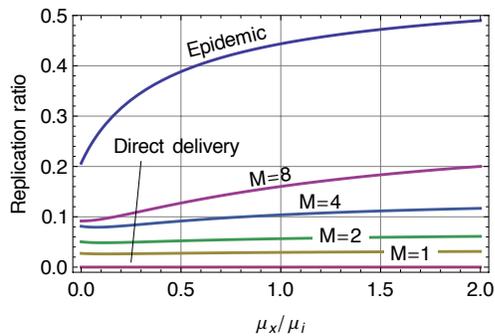
5. PERFORMANCE ANALYSIS

In this section, we evaluate performance of SAS and compare it with direct delivery search and epidemic search. Without lack of generality, we set $\mu_i = 1$ (scale of time) and vary the other parameters. In direct delivery search, n_s searches by itself with no assistance from the network, whereas in epidemic search all nodes with a replica of the query replicate the query to other nodes without being limited to a certain number of replicas. Obviously, these two schemes represent the two extremes, direct delivery having the least search cost and the epidemic leading the highest search performance in terms of response time. In the following, we set $n_1 = 8$ and $n_2 = 8$.

Fig. 3(a) shows the effect of μ_x on search time for various values of M . Initially, there are no seekers ($k_1 = 0$ and $k_2 = 0$) only one node in C_1 has the content. The y-axis represents the mean number of meetings n_s experiences before it receives a valid response.



(a) Response time.



(b) Search cost.

Figure 3: Effect of inter-community meeting time μ_x with various number of copies of the query M . Only one node in C_1 holds the searched content, i.e., $m_1 = 1$ and $m_2 = 0$, and $n_1 = 8$, $n_2 = 8$.

That is, it represents the search completion time. When $M = 0$, the search corresponds to the direct delivery, and with our assumptions on the mobility and the inter-meeting times, we have $\tau \sim \text{Exp}(m_1\mu_i + m_2\mu_x)$, with the mean $E[\tau] = 1/(m_1\mu_i + m_2\mu_x)$, or in terms of the meeting events, with aid of (1), we have

$$\frac{n_1\mu_i + n_2\mu_x}{m_1\mu_i + m_2\mu_x}.$$

In agreement with the previous works on direct delivery protocols, our result also shows the inefficiency of this search scheme. It has the highest search time. Employing one node as seeker, i.e. setting $M = 1$, decreases the time for search compared to the direct delivery search. The epidemic search represents the upper bound in search performance. With increasing M , the performance of SAS improves and approaches that of the epidemic search. This is not surprising as more nodes are aware of the search, more assistance they can provide. However, it should be noted that after some point, e.g. $M \approx 4$, the increase in M improves the search performance only marginally. Therefore, M can be set, e.g., to 4 in this search setting. *Replication ratio*, the ratio of the nodes that obtain the searched content replica to the whole population n , represents the cost of the search. In other words, the replication ratio is the expected fraction of nodes being tagged seekers. Fig. 3(b) compares the cost of search in case of SAS with various M values, direct delivery, and epidemic search. As can be seen from the figure, epidemic search spreads the content to almost half of the nodes in the network while SAS keeps the spread controlled via M parameter. Considering these two figures, we can say that SAS provides a bal-

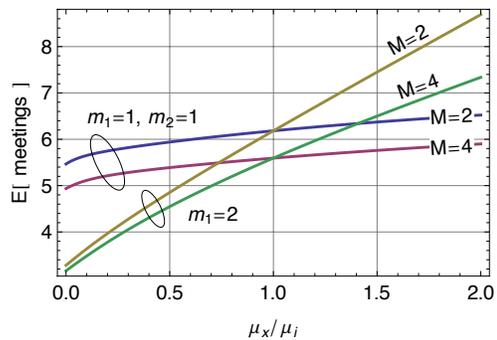


Figure 4: Mean search time when both C_1 and C_2 hold one copy of the searched data, i.e., $m_1 = 1$ and $m_2 = 1$.

ance between the search response time and cost. If we also add the cost of query replications, which is missing in replication ratio, the cost of epidemic search would be even much higher.

As the ratio μ_x/μ_i increases, the inter-community meetings occur more frequently compared to intra-community meetings. At the same time, the response time increases, especially with SAS. As epidemic search replicates data to every node, it is not affected significantly in terms of response time. On the other hand, cost of search with SAS is almost insensitive to the changes in μ_x/μ_i due to the activity being pushed to the right community, C_1 . As epidemic search does not make any differentiation among communities, it wastes the resources by replicating the query and content unnecessarily.

Fig. 4 compares the scenario in which both C_1 and C_2 have one copy of the searched content with the scenario in which C_1 has two copies, i.e., $m_1 = 2$ and $m_2 = 0$. For $\mu_x/\mu_i < 1$, which is the expected operating region according to the notion of community, intra-community meetings are more dominant. Therefore, the search takes shorter when C_1 holds both copies. However, as inter-community meetings become more frequent, the initial location of the content becomes less important. For $\mu_x/\mu_i > 1$, the social interactions among nodes somehow contradict with the notion of community and some other search strategy might actually fit better to this operating region.

Fig. 5 shows mean search time when the content is initially only in the other community C_2 . In this case, for low values of μ_x/μ_i , reaching the content unavoidably takes a long time, as the two communities rarely meet. However, for larger values of μ_x/μ_i , search speed increases significantly. E.g., for $\mu_x/\mu_i > 0.5$, the initial location of content does not seem to matter so much anymore. Finally, Fig. 6 illustrates the effect of number of nodes in the network to the mean response time. With the increasing network population, and a fixed number of initially tagged nodes ($m_1 = 1$ and $m_2 = 0$), the response time scales approximately linearly.

6. CONCLUSIONS AND FUTURE WORKS

Search without an infrastructure support in mobile clouds is non-trivial due to various dynamics of the system: highly dynamic topology of the network, constantly changing connectivity among nodes, and the lack of central information database. We proposed a search scheme—*seeker-assisted search*—in which the searching node spreads a limited number of copies of the query to the nodes it meets. These so-called seekers, get instructions about the query and thereafter search on behalf of the searching node. Our numerical evaluation suggests that such a seeker-assisted search decreases

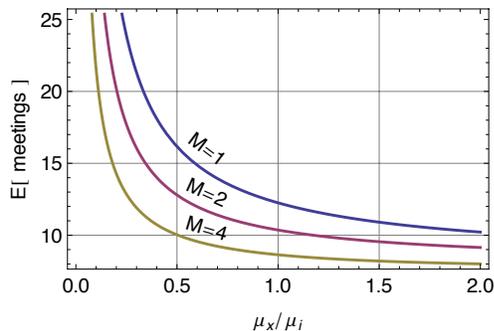


Figure 5: Mean search time when only C_2 has the searched data, i.e., $m_1 = 0$ and $m_2 = 1$.

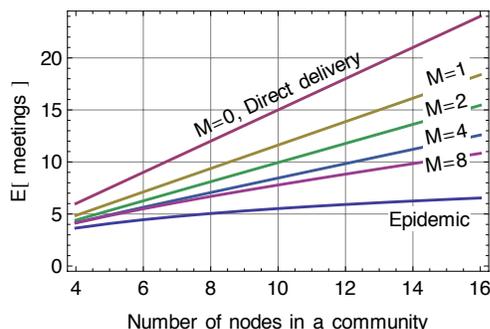


Figure 6: Mean search time with increasing network population for $m_1 = 1$, $m_2 = 0$, $\mu_x = 0.5\mu_i$.

the search cost compared to epidemic search and also increases the search performance compared to the direct delivery search.

For the sake of tractability, we assumed that pairwise inter-contact times are exponentially distributed. In our future work, we will loosen this assumption and devise a general model, including also scenarios with spatial correlations and lower degree of mobility. We will also go beyond model-based evaluation towards simulation studies with different inter-contact time distributions, using both synthetic mobility models and contact traces.

We believe that the following points deserve further attention for designing a complete search scheme in mobile opportunistic networks: our scheme differentiates nodes only according to their communities due to our assumption that the searched content is more probable to exist in the same community as the searching node.² However, the utility of each node within a community in relaying a query can be different. Hence, selecting the nodes with higher utility as seekers improves the search performance. In such a scheme, the nodes with higher interest similarity with the query would be preferred in assisting the search process. We plan to work on characterising nodes according to their interest profiles automatically from their stored contents.

7. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland in the PDP project with grant number 260389.

²We assume that nodes can easily be associated with a community as a Boolean decision.

8. REFERENCES

- [1] Eyuphan Bulut and B Szymanski. Exploiting friendship relations for efficient routing in mobile social networks. *IEEE Trans. on Parallel and Dist. Sys.*, 12:2254–65, 2012.
- [2] Han Cai and Do Young Eun. Crossing over the bounded domain: from exponential to power-law intermeeting time in mobile ad hoc networks. *IEEE/ACM Trans. on Networking (TON)*, 17(5):1578–1591, 2009.
- [3] Jialu Fan, Jiming Chen, Yuan Du, Ping Wang, and Youxian Sun. Delque: A socially aware delegation query scheme in delay-tolerant networks. *IEEE Trans. on Vehicular Technology*, 60(5):2181–2193, 2011.
- [4] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.
- [5] Wei Gao and Qinghua Li. Wakeup scheduling for energy-efficient communication in opportunistic mobile networks. In *IEEE INFOCOM*, 2013.
- [6] Pan Hui, Jeremie Leguay, Jon Crowcroft, James Scott, Timur Friedman, and Vania Conan. Osmosis in Pocket Switched Networks. In *IEEE First Int. Conf. on Comms. and Netw. in China*, 2006.
- [7] Pan Hui and Eiko Yoneki. BUBBLE Rap : Social-based Forwarding in Delay Tolerant Networks. *IEEE Trans. Mobile Comp.*, pages 1–14, 2011.
- [8] Pan Hui, Eiko Yoneki, Shu Yan Chan, and Jon Crowcroft. Distributed community detection in delay tolerant networks. In *Proc. of 2nd ACM/IEEE Int. Workshop on Mobility in the evolving internet architecture*, 2007.
- [9] Thomas Karagiannis, Jean-yves Le Boudec, and Milan Vojnovi. Power Law and Exponential Decay of Intercontact Times between Mobile Devices. *IEEE Trans. on Mobile Computing*, 9(10):1377–1390, Oct. 2010.
- [10] Vincent Lenders, Gunnar Karlsson, and Martin May. Wireless ad hoc podcasting. In *IEEE Conf. on Sensor, Mesh and Ad Hoc Comms. and Netws.*, 2007.
- [11] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.
- [12] Abderrahmen Mtibaa, Martin May, Christophe Diot, and Mostafa Ammar. Peoplerrank: Social opportunistic forwarding. In *Proc. of IEEE INFOCOM*, 2010.
- [13] Nam P Nguyen, Thang N Dinh, Sindhura Tokala, and My T Thai. Overlapping communities in dynamic networks: their detection and mobile applications. In *Proc. of the 17th Int. Conf. on Mobile Comp. and netw.*, pages 85–96, 2011.
- [14] Andrea Passarella and Marco Conti. Analysis of individual pair and aggregate inter-contact times in heterogeneous opportunistic networks. *IEEE Trans. on Mobile Comp.*, 2012.
- [15] Mikko Pitkänen, Teemu Kärkkäinen, Janico Greifenberg, and Jörg Ott. Searching for content in mobile DTNs. In *IEEE Int. Conf. on Pervasive Computing and Comms.*, 2009.
- [16] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proc. of the ACM SIGCOMM workshop on DTN*, 2005.
- [17] Ying Zhu, Bin Xu, Xinghua Shi, and Yu Wang. A survey of social-based routing in delay tolerant networks: Positive and negative social effects. 15(1), First quarter 2013.
- [18] Gjergji Zyba, Geoffrey M Voelker, Stratis Ioannidis, and Christophe Diot. Dissemination in opportunistic mobile ad-hoc networks: The power of the crowd. In *Prof. of IEEE INFOCOM*, pages 1179–1187, 2011.