A Socially Inspired Peer-to-Peer Resource Discovery Service for Delay Tolerant Networks

Tuan Dung Nguyen and Siegfried Rouvrais

GET / ENST Bretagne
{td.nguyen,siegfried.rouvrais}@enst-bretagne.fr

Abstract. The increasing popularity of wireless computing devices has promised a vision for mobile resource sharing applications. The scalability of such environments and their intermittent connection characteristics raise new challenges for both network protocols and system design. This paper proposes an overlay-based resource discovery service with a socially inspired peer-to-peer lookup algorithm for delay tolerant networks. Several simulation scenarios have been carried out to evaluate the algorithm's efficiency and scalability in comparison to classical approaches.

1 Introduction

The recent years have seen a remarkable diffusion of mobile appliances into our daily life. Nowadays, people often go around with their favorite handheld computing devices such as smart phones, PDAs or music players. Apart from their increasing processing and storage capacity, these devices can communicate via mobile ad hoc networks (MANETs) created on-the-fly using any available wireless interfaces. Many spontaneous applications [1] can be imagined that enable resource sharing among users anytime and anywhere (e.g. in the street, on the campus, at the airport).

However, spontaneous environments are often characterized by a large-scale population and intermittent connections between nodes due to their frequent mobility. Moreover, users can temporarily switch off their devices due to battery shortage. The end-to-end connectivity assumption in traditional MANETs does not hold all the time as nodes may be temporarily located in different network partitions. Delay tolerant [2] paradigms have been recently proposed to cope with the above-mentioned problems in which network protocols and system services should exploit node's physical mobility for message transmission with an acceptable delay.

So far, research works have primarily focused on routing issues to find a way to send a message from one node to another in these challenged networks [3]. But to the best of our knowledge, few relevant middleware services (e.g. naming, resource discovery) exist today to facilitate the construction of applications in these environments. For instance, the aim of resource discovery is to find the location of available resources (e.g. document, data) on the network. In traditional distributed systems, this service is often based on centralized directories and does not perform well in decentralized environment. Consequently, it is of interest to investigate peer-to-peer resource discovery techniques [4].

A scalable peer-to-peer resource lookup technique should avoid blind flooding in the network as it generates a lot of redundant messages leading to network congestion and battery wasting [5]. Furthermore, coping with intermittent connections is crucial to successfully provide relevant resources to their requesters. Our contribution is a socially-inspired resource lookup and delivery service that answers the aforementioned requirements. To the best of our knowledge, this is the first work composing several overlays for resource lookup in intermittently connected networks.

The rest of this paper is organized as follows. First, section 2 presents existing resource lookup and delivery approaches. Section 3 proposes our algorithm relying on two overlays of resource interest and human mobility. Section 4 evaluates and compares the performance and scalability of our approach with classical algorithms through simulations. Section 5 reviews the related work. Finally, section 6 concludes the paper and gives some perspectives for future work.

2 Background

We review in this section existing solutions for resource discovery in general and then clarify challenges related to inherent characteristics of delay tolerant networks. We discuss the benefits as well as issues relating to the composing of two overlays based on resource interest and human mobility.

2.1 Peer-to-peer resource lookup and delivery

Nodes are connected by logical links in an overlay network on top of underlying physical networks. A structured lookup technique (e.g. Chord, Pastry) imposes strict constraints with the overlay formation while an unstructured lookup technique (e.g. Gnutella) lets nodes self-organize to fulfill this task [4]. The former can achieve a more efficient resource lookup but also incur costly overhead due to the structured overlay's maintenance. The latter is believed to be better suited for coping with frequent topology changes and intermittent connections [6]. Additionally, unstructured lookup techniques can support better keyword searches.

Existing resource lookup mechanisms can be classified into three main categories: push-based, pull-based and combined push/pull [4]. Push-based solutions proactively broadcast resource advertisements in the networks. Nodes cache this information for their local lookup operations. This approach can also be used to disseminate small size spatial temporal resources. Pull-based solutions broadcast requests for resources on demand. Nodes having the requested resources answer with a reply message. This message can also be cached for later use to reduce the communication overhead. Other hybrid solutions combine the two previous push/pull approaches by broadcasting resource requests as well as advertisements. Push-based and push/pull based solutions are not relevant to non spatial temporal resources due to their generated communication overhead. On the other hand, pull-based solutions generate queries on demand and exploit the information from their propagation to send back reply message, e.g. using source routing.

Benefits of a resource interest overlay. Empirical studies of popular file-sharing systems (e.g. eDonkey) [7] have clearly demonstrated the interest proximity of resources offered by peers. Peers already having some common resources are likely to have others in the same category. Interestingly, this observation holds not only for popular but also for rare resources. Some works [7] have exploited this to significantly improve the lookup performance of Internet peer-to-peer file-sharing systems. A resource interest overlay can be obtained either by using peer's explicit profiles (peer's interest, sharing resource's categories) or implicitly investigating the lookup history. A peer tries to contact with his resource interest neighbors before soliciting other peers in the conventional overlay network. However, in networks with frequent mobility, a node may be unable to communicate directly with its corresponding resource interest neighbors.

2.2 Issues with delay tolerant networks

For discovery service, a simple broadcast can fail to reach relevant nodes due to mobility and intermittent connectivity. To cope with this issue, periodic broadcast or gossip-based (epidemic-style) [8] approaches should be taken into consideration. The former repeatedly diffuses messages to every 1-hop neighbors. The latter stores received messages in a buffer and forward them later a defined number of times t. Each time, a message is sent to a defined number f of randomly selected nodes. Additionally, frequent mobility often leads to the invalid reverse path. The return of reply message using source routing results in high latency or even failed delivery. On the other hand, a simple use of delay tolerant routing protocols (e.g. epidemic-routing) can be costly in term of high communication overhead.

Benefits of a human contact overlay. Experimental results [9] have demonstrated that human mobility is not totally arbitrary. The *small world* phenomena [10] suggested that two random U.S. citizens could be connected by hand-passing a letter by a short chain of six acquaintances. In reality, human movements are solicited by various social relations (e.g. family, workplace) [11]. Mobility-aided techniques, where messages can be transmitted thanks to node physical movements, have exploited this result to resolve routing problems in intermittently connected networks [3]. Likewise, these nodes can also be considered as good forwarding candidates in resource discovery mechanisms. A human contact overlay can be built by monitoring the meeting frequency with node's physical neighbors.

2.3 Composing two overlays

Keeping an updated global view of these overlays at each mobile nodes is an unfeasible task due to limited resources and frequent mobility. Therefore, desired global behaviors should be obtained from aggregated local information using appropriate rules without a complete system view. In our proposal, a node only keeps information about its neighbors in each overlay and obtains information about other nodes during lookup operations. Moreover, as depicted in Figure 1, nodes appearing as neighbors in one overlay may not be neighbors in the other. The inconsistency between two overlays O_1 and O_2 and node's physical neighbors raises difficulties for lookup operations and requires an algorithm capable of working on these two superposed overlays.

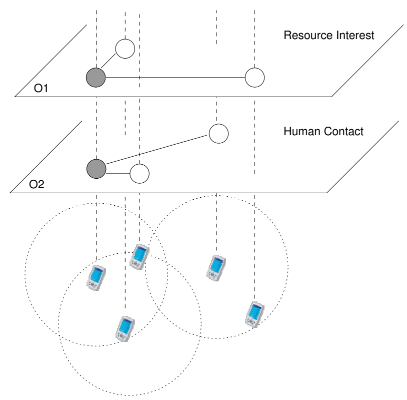

Fig. 1. Two overlays vs. physical topology

3 Socially-inspired resource discovery

3.1 System Model

The network is modeled by a graph $G = (V, E)$, where $V = p_1, p_2, \ldots, p_n$ is the set of nodes and E is the set of direct connections among them. The network dynamism implies that V and E evolve over time as nodes can be switched off or out of transmission range. A node has a list of physical neighbors currently in its transmission range. For our two selected overlays, we propose that each node maintains two semantic neighbor lists defined as follows:

1. **Interest Neighbor**. Resources are classified into a limited number of categories (e.g. music, events). Node keeps a list of nodes sharing the same interest, namely *interestNeighbor*, which are likely to provide resources in a category. The Least Recently Used (LRU) strategy [7] is used to maintain these lists with nodes having already replied to previous queries in the corresponding category.

2. **Contact Neighbor**. A node keeps a list of neighbor nodes in its social networks, namely *contactNeighbor*, i.e. they are more likely to be met in the future. Items in this list are in the form of $\{(p_i, timer)\}$ where *timer* represents the meeting frequency with p_i. The value *timer* is initially set to 0 and incremented by 1 at each meeting time.

3.2 Algorithm

Our algorithm is strongly inspired from human being's behaviors. An individual usually has some social relations with several *acquaintances* (e.g. family members, friends, colleagues). To look for a piece of information, one often starts by contacting as soon as possible those that he/she believes to be more likely to have a favorable answer. In case of unsuccess, he/she may then lean on his/her *acquaintances* to help find that information.

```
initialisation:
begin
    interestNeighbor ← ∅;
    contactNeighbor ← ∅;
    msgBuffer ← ∅;
end
while true do
    wait until event;
    switch event do
        case LookupEvent
            msg = CreateReqMessage ();
            for p ∈ physicalNeighbor do  send (p, msg);
        case RequestReceivedEvent (msg)
            if IsResourceFound (msg) then
                SendResponse ();
            else if msg.src ∈ contactNeighbor then
                msg.interestNeighbor ← msg.interestNeighbor ∪ interestNeighbor;
                msgBuffer ← msgBuffer ∪ msg;
        case GossipTimerEvent
            CGossip ();
            Update (contactNeighbor);
        case ResponseReceivedEvent (msg)
            Update (interestNeighbor);

```

Fig. 2. Main Lookup Algorithm

The main algorithm (cf. Figure 2) remains idle until it is triggered by one of these four principal events:

– LookupEvent: event generated by applications when they want to look for a resource;
– RequestReceivedEvent: event generated by the communication layer to inform its higher layers about a request message arrival;
– GossipTimerEvent: event generated periodically to trigger a gossip round;

– ResponseReceivedEvent: event generated by the communication layer to inform its higher layers about a response message arrival.

For the sake of clarity, Figure 2 illustrates the pseudo code of the main algorithm for one resource category. The query propagation works as follows. The generated query messages are first sent to any available corresponding interest neighbors. If this step fails, query messages are sent to every nodes in the transmission range and the message is augmented with the source node's interest neighbors (cf. line 16). This allows nodes to discover gradually other neighbors in the resource interest overlay from their initial local knowledge. Node receiving a query and having the required resource will send back a reply message to the query's origin. Otherwise, it saves a copy of the query in the message buffer and gossips it using the previously described mechanism. The gossiping algorithm is depicted in Figure 3. Furthermore, the lookup process for distant resources is carried out through a stable path consisting of social-related neighbors. The return of reply messages is realized using source routing to exploit the social traces left by the propagation of query messages.

```
Procedure CGossip ()
begin
    for msg ∈ msgBuffer do
        if Timeout (msg) then
            msgBuffer ← msgBuffer \ msg;
        else
            for p ∈ neighbor do
                if p ∈ msg.interestNeighbor then
                    send (p, msg);
                else
                    for p ∈ physicalNeighbor do  send (p, msg);
end
```

Fig. 3. Controlled Gossip Algorithm

4 Evaluation

4.1 Simulation environment

We realized our experimental study using OMNet++ [12], a modular simulation environment. We focused on the sent messages overhead of different decentralized lookup algorithms. We did not consider message loss nor low layer's detail in this study. Messages are supposed to be transmitted between nodes in transmission range with a random delay uniformly selected in the range $[0.1, 0.4]$ (s). The main simulation parameters are presented in Figure 4. During the simulations, nodes move following two mobility models: Random Way Point (RWP) and Community-based (CMM). The former is a classic model in which each node chooses a random target location, moves to the destination with a random speed, then waits for a random period of time before repeating this process. The latter is a recently proposed model that exploits social networks to generate more realistic mobility traces [11].

Parameters	Values	Parameters	Values
Simulation time	1800 s	Transmission delay	Uniform in $[0.1, 0.4]$ s
Simulation area	2000 m $\times$ 2000 m	Node speed	Uniform in $[1, 6]$ m/s
Transmission range	50 m	Request interval	120 s

Fig. 4. Simulation parameters

We evaluated our proposal by comparing it with two classical algorithms: *periodic broadcast* and *epidemic-based*. In the former one, node keeps its own queries in a buffer to periodically broadcast them to its neighbors. In the latter one, node does not only keep its own queries but also received queries in a buffer to periodically send them to a random subset of its current neighbors.

4.2 Resources and queries distribution model

A realistic workload is crucial for accurate algorithm evaluation. We assume that resources inspire from previously discovered characteristics of peer-to-peer file sharing applications: semantic proximity and Zipf-like distribution[7]. As no real workload exists so far in mobile environments, a synthetic model is used to generate resource distribution respecting the aforementioned characteristics.

We adopted the Number Intervals model used in [13] in which a resource is represented by a point in the interval $[0, 1]$ and a query is represented by a range within that interval, e.g. $[0.2, 0.5]$. A query is hit when there is a resource falling into its range. We limited our simulation with 10 categories of resources represented by 10 intervals $[0, 0.1], [0.1, 0.2], \ldots, [0.9, 1]$. Each node is also limited with one semantic category. Inspired from experimental results in [7], we proposed that a resource is generated in its corresponding semantic nodes with a probability equals to 0.7.

As in [13], a resource query is also represented by a range $(center, range)$ where *range* takes a random value according to a normal distribution with mean 0.05 and variance 0.002 whereas *center* is selected according to a Zipf-like distribution. The probability that *center* falls into a category i is $\frac{(\frac{1}{i})}{\sum_{j=1}^{10}(\frac{1}{j})}$.

4.3 Simulation results and analysis

The presented algorithms are evaluated using the three following criteria: *success ratio* (the number of successful resource deliveries/the number of resource requests) and *total sent messages*.

The hit ratio is depicted in Fig. 5 (a) and the message overhead is illustrated in Fig. 5 (b) with the message overhead in log scale. The periodic broadcast and epidemic algorithms trade off message overhead for hit ratio. Our algorithm achieved a hit ratio as good as other algorithms while reducing message overhead by order of magnitude.

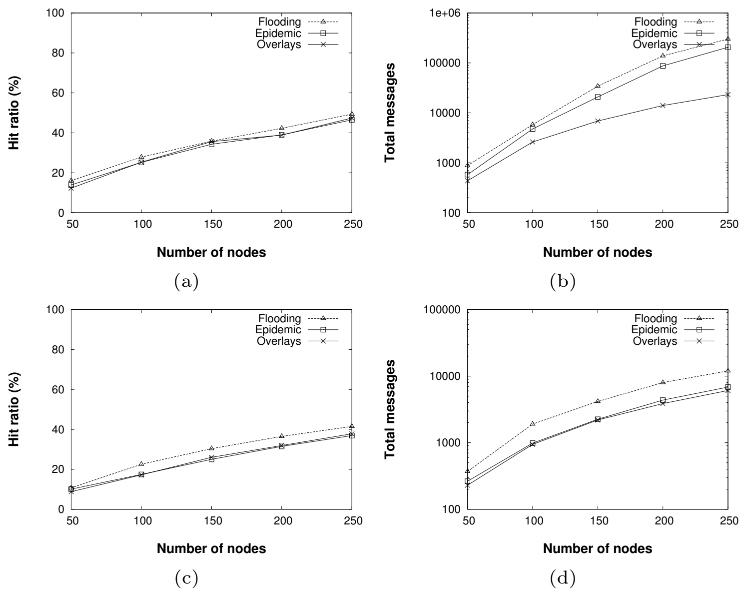

Fig. 5. Hit ratio vs. message overhead: (a), (b) with CMM model; (c), (d) with RWP model

Impact of node density. We ran our simulations with 50, 100, 150, 200, 250 nodes to verify the impact of node density to the algorithm's performance and scalability. A smaller number of nodes results in a sparser network that is closer to our application's environment. The simulation results clearly show that our proposed algorithm is really appropriate for intermittently connected networks.
Impact of mobility model. The simulation results in Fig. 5 (c) and Fig. 5 (d) show that our algorithm works considerably better than other algorithms with the CMM model, i.e. with social-related human mobility. However, its performance is not worse than the other algorithms with the RWP model where a node's movement is completely random.

5 Related work

Several works have been proposed for lookup in highly dynamic environments using push, pull or push/pull approaches and caching mechanisms. None of them have taken into account the coexistence of multiple overlays for representing non-functional contexts.

Lindemann et al. [14] proposed an epidemic-based peer-to-peer lookup service, namely Passive Distributed Indexing (PDI), to cope with intermittent connectivity and high mobility. Query and response messages are transmitted using local broadcast and query results are cached in participating nodes for later use. Most queries could be resolved locally thanks to the implicit dissemination of index entries in the network using node mobility. This work uses a pull-based approach as ours but does not take into account content semantic proximity and social-related mobility.

Motani et al. [15] presented the PeopleNet architecture for searching information in a wireless virtual social network. Information queries which represent both requests and advertisements are first directed via infrastructure-based networks to k randomly selected nodes in non overlapping geographical areas and then propagated in a peer-to-peer manner. The key idea is that a better probability of match (i.e. information found) is achieved when related matching queries are closely placed in a defined area. Unlike our work, this work uses a combined push/pull approach and also needs an infrastructure-based network.

Wolfson et al. [13] proposed an algorithm called Rank-Based Broadcast (RBB) for the discovery of local spatio-temporal resources in high mobility environments. This work follows a combined push/pull approach where both resource information (namely reports) and queries are disseminated in the network by periodic 1-hop broadcast Reports are ranked by their relevancy to queries received from other nodes and only the top-ranked ones are sent in each broadcast round. However, this algorithm generates high communication overhead with large size resources and its lookup efficiency can be degraded with rare resources.

Hui et al.[16] presented a new communication scheme, namely Osmosis, for file sharing in Pocket Switched Networks (PSN). Lookup messages are disseminated using an epidemic scheme while a penalty-based osmosis scheme is proposed to send back reply messages with a certain level of reliability and without overloading the network. Reply messages are implicitly directed to lookup senders using traces left during the lookup process due to the propagation of query messages. This work is inspired from a biological phenomenon and does not exploit social and resource semantic as in our work.

6 Conclusion

We have proposed a socially-inspired algorithm for resource lookup and delivery in delay tolerant networks. We have demonstrated through simulations that our algorithm can achieve as good success ratio as existing solutions while considerably reducing the communication overhead and latency. We plan to carry out further evaluations with better metrics to determine more accurate interest and contact neighbors as well as with a a real resource workload. Future work will also consist of proposing a middleware to take into account different contexts (e.g. battery, reputation, application's non-functional requirements) and to generalize our multiple overlays approach.

References

1. Feeney, L.M., Ahlgren, B., Westerlund, A.: Spontaneous Networking: An Application-Oriented Approach to Ad Hoc Networking. IEEE Communications Magazine **39**(6) (2001) 176–181
2. Burleigh, S., Hooke, A., Togerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., Weiss, H.: Delay-Tolerant Networking: An Approach to Interplanetary Internet. IEEE Communications Magazine **41**(6) (2003) 128–136
3. Pelusi, L., Passarella, A., Conti, M.: Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. IEEE Comm. Mag. **44**(11) (2006)
4. Li, X., Wu, J.: Searching Techiques in Peer-to-Peer Networks. In Wu, J., ed.: Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks. Auerbach Publications, Boston, MA, USA (2005) 613–642
5. Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The Broadcast Storm Problem in a Mobile Ad Hoc Network. In: Proc. of MobiCom'99, (Seattle, Washington, USA)
6. Oliveira, L.B., Siqueira, I.G., Macedo, D.F., Loureiro, A.A.F., Wong, H.C., Nogueira, J.M.: Evaluation of Peer-to-Peer Network Content Discovery Techniques over Mobile Ad Hoc Networks. In: Proc. 6th IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'05), Taormina, Italy (2005)
7. Handurukande, S.B., Kermarrec, A.M., Fessant, F.L., Massoulié, L., Patarin, S.: Peer Sharing Behaviour in the eDonkey Network, and Implications for the Design of Server-less File Sharing Systems. In: Proc. 1st European Conf. on Computer Systems (EuroSys'06), Leuven, Belgium, ACM Press (2006)
8. Eugster, P.T., Guerraoui, R., Kermarrec, A.M., Massoulié, L.: Epidemic Information Dissemination in Distributed Systems. IEEE Computer **37**(5) (2004) 60–67
9. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms. In: Proc. of INFOCOM'06, Barcelona, Spain (2006)
10. Milgram, S.: The Small-World Problem. Psychology Today (1967) 61–67
11. Musolesi, M., Mascolo, C.: A Community Based Mobility Model for Ad Hoc Network Research. In: Proc. 2nd ACM/SIGMOBILE Int. Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN'06), (Florence, Italy) 31–38
12. OMNeT++. `http://www.omnetpp.org` (2006)
13. Wolfson, O., Xu, B., Yin, H., Cao, H.: Search-and-Discover in Mobile P2P Network Databases. In: Proc. 26th IEEE Int. Conf. on Distributed Computing Systems (ICDCS'06), Lisboa, Portugal, IEEE CS Press (2006)
14. Lindemann, C., Waldhorst, O.P.: A Distributed Search Service for Peer-to-Peer File Sharing in Mobile Applications. In: Proc. 2nd IEEE Int. Conf. on Peer-to-Peer Computing (P2P'02), Linköping, Sweden (2002)
15. Motani, M., Srinivasan, V., Nuggehalli, P.S.: PeopleNet: Engineering A Wireless Virtual Social Network. In: Proc. 11th ACM Int. Conf. on Mobile Computing and Networking (MOBICOM'05), Cologne, Germany, ACM Press (2005) 243–257
16. Hui, P., Leguay, J., Crowcroft, J., Scott, J., Friedman, T., Conan, V.: Osmosis in Pocket Switched Networks. In: Proc. 1st Int. Conf. on Communications and Networking in China (CHINACOM'06), Beijing, China (2006)