Enabling a Mobile Cloud Service: Data-Sharing in Ad-hoc Device-to-Device Mobile Networks

Kavitha Ranganathan

W. P. No. 2017-03-04
March 2017

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD-380 015
INDIA

Enabling a Mobile Cloud Service: Data-Sharing in Ad-hoc Device-to-Device Mobile Networks

Kavitha Ranganathan
Indian Institute of Management, Ahmedabad

Abstract

The objective of this work is to build a data-sharing application for an ad-hoc network of mobile devices, where users can exchange data/files among themselves without relying on traditional communication channels like telecom or network operators. In other words, we aim to build a mobile cloud service for data sharing. This paper examines the resource discovery and selection (also called replica selection) issue in such a mobile cloud. We propose a novel decentralized algorithm where nodes can first discover and then choose the best replica to request for, from among the different alternatives identified. Additionally, our paper comes up with a new metric to evaluate different replicas, that is, what could be a desirable definition of the 'best' replica in such a network.

Keywords: mobile ad-hoc network, device-to-device, data-grid

1. Introduction

Ad hoc device-to-device mobile phone networks have recently been proposed as an alternate means for data sharing. In these networks, specially designed mobile phones can directly communicate with one another. More interestingly, these phones can act as intermediaries – routing data (audio or otherwise) between two phones that are out of range from each other. Hence, a collection of such phones can form an ad hoc network among themselves. For low-income groups or special interest groups, this technology provides an alternate and cheap mode of both telephony and file sharing.

The objective of our work is to build a file-sharing application for such ad-hoc networks where users can easily exchange data/files among themselves at zero network cost – hence building a mobile cloud service for data sharing. This novel application layer brings up a number of distributed file sharing issues related to resource discovery, caching, replication, incentive structures and network utilization. This paper specifically looks at the resource discovery and selection aspects of file-sharing in such a mobile cloud.

Since the entire network is totally decentralized, it is imperative that the solutions to resource-discovery also operate in a decentralized fashion. Since multiple replicas of a certain file may exist in the network, the node requesting a file should be able to choose the best replica. The definition of best in this context would mean the replica that has the highest chance of being transferred to the requester speedily. Various factors could play a role here including route stability, number of hops and network congestion.

Our paper proposes a novel decentralized algorithm where a requesting node is informed about all possible replicas and their relative rating for suitability for download. This is achieved by nodes maintaining one-hop knowledge of their neighbors and hence the stability of the one-hop routes around them, which is incrementally built to calculate the suitability of various routes and hence the different replicas.

The rest of the paper is organized as follows: the next section contains a review of relevant literature and section 3 elaborates on the ad hoc network model and the data-sharing application which layers on top of it. In Section 4 we detail the proposed resource discovery and selection algorithm. We conclude in Section 5 along with directions for future work.

2. Related Work

Past work has looked at using mobile ad hoc networks for a range of applications including communication, information dissemination and data sharing. The application this paper is concerned with is data sharing, and this section concentrates on related work in this domain. Very limited work that specifically addresses resource discovery and replica selection for this particular application exists.

To address ad hoc peer-to-peer data dissemination, Goel et al (Goel et al., 2002) suggest applying Toronto Coding to the problem. In their proposed solution, parts of a file are coded and saved among different peers. When needed, these different parts can be downloaded from various peers and the file can be reconstructed. This presumably leads to better performance as dependency on one node for the entire file is minimized.

The REDMAN middleware (Bellavista et al., 2005) is a solution for disseminating and retrieving replicas in a Manet. Most of their work concentrates on calculating and creating the optimal number of replicas and ensuring availability of files. For file retrieval, they propose using designated nodes as 'resource delegates', who will have knowledge about replica locations. While their solution does not use a centralized repository, it does delegate extra work to certain nodes in the network.

Some work in resource retrieval in Manets resort to positioning systems like GPS (Aydin and Shen, 2002) while others impose restrictions on where replicas can be placed, to ensure quick retrieval . For example Tamori et al. (Tamori et al., 2002), propose that a node be allowed to create replicas only within n hops of itself, thus exploiting this feature when a replica needs to be retrieved.

Studies on Data sharing and caching in Manets (Cao et al., 2004; Du et al., 2009; Fan et al., 2013; Mershad and Artail, 2011), look at multiple perspectives. Cao et al. look at building a cooperative cache, by replication data along the routes joining clients and servers in an ad hoc network. Du et al. propose using cooperative zones in which data is cached. This they claim ensures that there is less communication overheads on the data sources. Fan et al. propose a caching model that exploits temporal relationships between data items. Each node mains a progress report of data it uses and shares this with other nodes. These progress reports help nodes decide whether and when to cache data items. Mershad et al. follow an approach where nodes cache fragments of earlier requests. When a new request comes in, nodes cooperate to stitch together various fragments to produce the answer.

3. Description of Network and Application Model

As discussed earlier, the network we consider consists of individual users with mobile devices that can connect directly to each other using a technology like IEEE 802.11 (Wifi). These devices also forward data to other nodes within their range and hence act as routers. A group of such devices form a Mobile Ad Hoc Network (MANET). These users could belong to different kinds of communities, for example students on a university campus or participants in a conference venue. The application we study in this paper, is a data-sharing service that any of these users in this network can be a part of. This is in effect a distributed data cloud, albeit for a limited local community of users.

Such a data-sharing service will entail that popular files in the community are stored at multiple devices, to avoid overload of specific nodes or routes. Hence, replica creation and placement, which is a well -researched topic in traditional peer-to-peer and grid systems would play an important role here as well.

When a node needs a particular resource (called a particular data file hence forth), it will first have to find out the locations or node ids where the replicas of that file reside. This process is referred to as the *resource discovery* phase. In traditional grid systems, resource discovery could be a simple process if a central repository of file locations is maintained. However, in a MANET, where we assume no centralized control or action, the resource discovery process cannot assume a central repository and has to be a decentralized and dynamic process.

Once a node has discovered multiple replica locations for the file required (note, it may or may not discover all the replicas of the file), it needs to decide which replica to request for. This process is called the *resource selection* phase.

Our proposed decentralized algorithm for resource discovery and selection is detailed in the next section.

4. Resource Discovery and Selection Algorithm

We propose an algorithm which has two phases: the resource discovery phase and resource selection phase. These phases are not strictly sequential but overlap to some degree. In fact, the resource selection algorithm piggy-backs on the resource discovery phase as detailed below.

Resource discovery : Let us assume node N wants a particular file (F1) from the data cloud. It initiates a search for the file by sending out a broadcast request for the file. Its neighbors receive the request and broadcast it to their neighbors and so on till a node (or nodes) that have F1 receive the request.

The algorithm described above is called flooding (Vollset, 2003), a common technique to spread a particular message to all the nodes in a MANET. However, flooding has been shown to be highly inefficient because of indiscriminate forwarding – especially in dense networks (Tseng et al., 2002). Other algorithms that are more efficient have been proposed in the past (Williams and Camp, 2002), and we use SBA (Scalable Broadcast Algorithm) (Wei Peng and Xi-Cheng Lu, 2000), which is known to perform well for many Manet scenarios. In SBA, a node (say N1) receiving the message waits for a predefined time interval called the RAD (Random Access Delay), and keeps track of duplicate messages received from its neighbors. If after the RAD times out, there are still some neighbors left who have not sent N1 the message, then N1 decides to forward the message, else it drops it. The basic premise of SBA is that nodes forward a message, only if anybody in their neighborhood has not yet got that message from someone else. SBA clearly brings down redundancy in message forwarding and hence lowers the overheads of flooding the network.

However, since the purpose of our broadcast is to file a copy of a particular file and not reach all nodes in the network, we propose a modified SBA, which we call the expanding ring SBA (inspired by the expanding ring search in AODV, a routing protocol (Perkins and Royer, 1997). The rational of the expanding ring SBA, is that a node will prefer to access replicas near it than many hops away.

Expanding ring SBA works as follows : a node uses SBA to broadcast its request to only k hops around it. The broadcast message is dropped after k hops. If a node within k hops has the file, it replies and the reply message follows the same route back to the requester. However, if no replicas exist within a radius of k hops, the requester, initiates another request for the same file, but this time the search radius is expanded to k+n hops and so on. The intuition behind the expanding ring search is that overheads on the network in terms of request messages are reduced albeit at added latency if the replicas are many hops away.

The reply message from a node keeps track of two additional metrics : the number of hops between replier and requester, and the stability rating of the route between the requester and replier. Both these metrics help the requester in the next phase when it has to choose the best replica.

5. Selection of Resource

When a node has to select the ‘best’ replica, there are two primary factors that need to be considered : the number of hops the data has to go through to reach the destination and the stability of the route.

Number of hops in route: The number of hops the data file has to traverse has bearings on the delay and latency involved in the file transfer.

Stability of route: In a MANET, since the network by definition is dynamic, links between nodes may vary in their stability. Hence, more stable routes may work better to ensure that the entire file (which will be transferred via multiple individual packets) is transferred from the source to the destination before one or more links in the route break.

Both these metrics are calculated in the reply message that the node sends back to the requester. Suppose node A has requested for a file, and the request goes via node B to node C. Node C, replies saying it has the file, and the reply travels via B, back to A.

We introduce the concept of the stability of a route, by first assuming that each hop in a route has a stability measure. How this stability measure is calculated for each hop is detailed in the next subsection. The cumulative amount of this measure for all the hops in a route is the stability rating of a route.

For example :

$\text{Stability}_{A:C} = \text{Stability}_{A:B} + \text{Stability}_{B:C}$ where $\text{Stability}_{N1:N2}$ denotes the stability of the link between N1 and N2.

As the reply message travels from C to A via B, C (which has kept track of the stability of all the links emanating from it), adds $\text{Stability}_{B:C}$ to the reply message. Once the message reaches B, B adds $\text{Stability}_{A:B}$ to the reply. When the reply reaches A, A has the complete stability rating for the entire route between itself and C.

Similarly, A would get stability ratings of all the routes to all the replicas of the file and can select the best one. The main advantage of the algorithm is that a node does not generate extra overheads to compile the stability of routes to different replicas. This information is piggy-backed on the reply message from the various sources, thus saving network bandwidth and avoiding additional delays.

6. Calculating the stability of a link

While we have made it clear how the end-to-end stability of a route is compiled, the issue of how a particular link's stability is calculated is explained below.

Past work has looked at predicting link stability in the context of routing algorithms. A number of solutions including link age (Toh, 1997), signal strength (Molnar and Marie, 2011) and affinity of neighbors (Agarwal et al., 2000) has been proposed. We adopt link age as a proxy for link stability as it has the least overheads and is straightforward to calculate. Each node maintains a neighbor list as part of the SBA protocol. To this neighbor list, we add a field that keeps track of how long each node has been in the list. This simple addition provides the metric needed for link stability.

Since there could be a tradeoff between the stability of a route versus the length of a route (number of hops), a metric that combines both these with specific weights could be used to decide which replica to choose. Note that as the number of hops in a route increase, this could lead to more latency and delay in getting the entire file. On the other hand, if the stability of a route decreases there probability of incomplete transfers and retries increases. This and other tradeoffs will be studied further when we test our algorithm in the simulations studies proposed for future work.

7. Conclusions and Future Work

Device-to-device data sharing services for a community of local users are mushrooming as possible applications for mobile clouds. The network substrate for such an application are mobile ad-hoc networks, which have traditionally been explored for communication in disaster relief or hostile environments which do not have traditional network coverage.

We have proposed a novel resource discovery and selection algorithm for such a data-sharing service. Our proposal is to compile and use link stability to help a node choose the most favorable replicas of the file needed. At the same time, our algorithm ensures that additional overheads in terms of network utilization are not incurred while compiling and propagating this information.

Going forward, we plan to use discrete-event simulations on a wide range of scenarios, to test the robustness and effectiveness of our proposed algorithm on a number of parameters like latency, bandwidth consumption, memory usage and other possible overheads. Additionally, the mobility models that have been studied for such networks are very limited (usually the random walk mobility model which has been proved in the past to being unsuitable for MANET studies). We propose to develop a new mobility model, suitable for the application in question : special interest groups or communities who want to create a mobile cloud among themselves for data sharing.

References

Agarwal, S., Ahuja, A., Singh, J.P., Shorey, R., 2000. Route-lifetime assessment based routing (RABR) protocol for mobile ad-hoc networks, in: 2000 IEEE International Conference on Communications, 2000. ICC 2000. Presented at the 2000 IEEE International Conference on Communications, 2000. ICC 2000, pp. 1697–1701 vol.3. doi:10.1109/ICC.2000.853783

Aydin, I., Shen, C.-C., 2002. Facilitating match-making service in ad hoc and sensor networks using pseudo quorum, in: Eleventh International Conference on Computer Communications and Networks, 2002. Proceedings. Presented at the Eleventh International Conference on Computer Communications and Networks, 2002. Proceedings, pp. 4–9. doi:10.1109/ICCCN.2002.1043037

Bellavista, P., Corradi, A., Magistretti, E., 2005. REDMAN: An optimistic replication middleware for read-only resources in dense MANETs. Pervasive Mob. Comput. 1, 279–310.

Cao, G., Yin, L., Das, C.R., 2004. Cooperative Cache-Based Data Access in Ad Hoc Networks. Computer 37, 32–39. doi:10.1109/MC.2004.1266293

Du, Y., Gupta, S.K.S., Varsamopoulos, G., 2009. Improving On-demand Data Access Efficiency in MANETs with Cooperative Caching. Ad Hoc Netw 7, 579–598. doi:10.1016/j.adhoc.2008.07.007

Fan, X., Cao, J., Mao, H., Liu, Y., 2013. Gossip-based Cooperative Caching for Mobile Applications in Mobile Wireless Networks. J Parallel Distrib Comput 73, 653–663. doi:10.1016/j.jpdc.2013.01.006

Goel, S.K., Singh, M., Xu, D., Li, B., 2002. Efficient Peer-to-Peer Data Dissemination in Mobile Ad-Hoc Networks.

Mershad, K., Artail, H., 2011. CODISC: Collaborative and Distributed Semantic Caching for Maximizing Cache Effectiveness in Wireless Networks. J Parallel Distrib Comput 71, 495–511. doi:10.1016/j.jpdc.2010.11.001

Molnar, M., Marie, R., 2011. Stability Oriented Routing in Mobile Ad Hoc Networks Based on Simple Automatons. Mob. Ad-Hoc Netw. Protoc. Des. 363–390.

Perkins, C.E., Royer, E.M., 1997. Ad-hoc On-Demand Distance Vector Routing, in: IN PROCEEDINGS OF THE 2ND IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS. pp. 90–100.

Tamori, M., Ishihara, S., Watanabe, T., Mizuno, T., 2002. A replica distribution method with consideration of the positions of mobile hosts on wireless ad-hoc networks, in: 22nd International Conference on Distributed Computing Systems Workshops, 2002. Proceedings. Presented at the 22nd International Conference on Distributed Computing Systems Workshops, 2002. Proceedings, pp. 331–335. doi:10.1109/ICDCSW.2002.1030790

Toh, C.-K., 1997. Associativity-Based Routing for Ad Hoc Mobile Networks. Wirel Commun 4, 103–139. doi:10.1023/A:1008812928561

Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., Sheu, J.-P., 2002. The broadcast storm problem in a mobile ad hoc network. Wirel. Netw. 8, 153–167. doi:http://dx.doi.org/10.1023/A:1013763825347

Vollset, E., 2003. A survey of reliable broadcast protocols for mobile ad-hoc networks.

Wei Peng, Xi-Cheng Lu, 2000. On the reduction of broadcast redundancy in mobile ad hoc networks, in: 2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. Presented at the 2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC, IEEE, pp. 129–130. doi:10.1109/MOBHOC.2000.869221

Williams, B., Camp, T., 2002. Comparison of broadcasting techniques for mobile ad hoc networks, in: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing - MobiHoc ’02. Presented at the the 3rd ACM international symposium, Lausanne, Switzerland, p. 194. doi:10.1145/513800.513825