# ComPAS: Maximizing Data Availability with Replication in Ad-hoc Social Networks

Ahmedin Mohammed Ahmed, Qiuyuan Yang, Nana Yaw Asabere, Tie Qiu, Feng Xia
School of Software, Dalian University of Technology
Dalian 116620, China
f.xia@acm.org

## ABSTRACT

Although existing replica allocation protocols perform well in most cases, some challenges still need to be addressed to further improve their performance. The success of such protocols for Ad-hoc Social Networks (ASNETs) depends on the performance of data accessibility and on the easy consistency management of available replica. We contribute to this line of research with replication protocol for maximizing availability of a data. Essentially, we propose ComPAS, a community-partitioning aware replica allocation method. Its goals include integration of social relationship for placing copy of the data in the community to achieve better efficiency and consistency by keeping the replica read cost, relocation cost and traffic as low as possible.

## Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General - *Data communications*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software - *Distributed systems*

## Keywords

Ad-hoc social network; data replication; middleware

## 1. INTRODUCTION

Ad-hoc social networks (ASNETs) are a branch of modern wireless networks or a combination of social networks and Mobile Ad-hoc Networks (MANETs) [1]. Data management in ASNETs is essential due to characteristics of nodes, such as mobility and resource constraints. This causes the nodes to make a social community in which data from two separated communities become inaccessible to each other. Ensuring data availability at the point of community-partitioning is a very significant issue. One of the key solution to increase data availability in ASNET is to replicate the data items in the community that are not the owners of original data.

However, the problem in ASNETs is, most of the operations are based on the data of a user and neighbors' data. Replicating user's data in multiple or all the servers eliminates the inter-server traffic for reads but increases the replication overhead [2]. In our case, we assume that; 1) a user belongs only to one community, and 2) created social communities should be determined not only by common interests or contacts but also by mobility-related context. Hence, nodes collaborate and move as a group instead of individually. Even though the ComPAS replica allocation method is very young, we have already obtained some remarkable results. We have shown that the method is generic enough to provide data availability in an efficient way. We have also succeeded in managing consistency of the efficiency and keeping the relocation cost as low as possible.


*WWW'14 Companion,* April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577319.

## 2. THE COMPAS PLATFORM

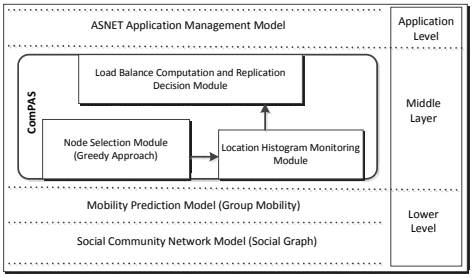


**Figure 1: ComPAS system model.**

The system model depicted in Fig. 1 presents ComPAS's operation which is shown below in Algorithm 1 and the connection it has with the lower and upper layers. Due to the mobility in ASNETs, nodes with similar moving behavior form a community. That is, $N$ user nodes with a set of $G$ communities. $D_{ic} = 1$ if and only if user $i$'s data is stored at one of the storage space at community $c$, and $\sum_{c=1}^{G} D_{ic} = 1 \forall i$. Our aim is to find an efficient and consistent way to store $X$ replicas for each user's data on the storage space at G communities ($X < G$). We choose the value of $X$ depending on the replication budget of the system and it's desired availability. We considered two type of data queries (read and write). The read query represents the number of communities required to retrieve the data of user $i$ and that of every neighbor of $i$. To retrieve user $i$'s data, the query is sent to its primary/own community, say in community $c$ (i.e., $D_{ic} = 1$), incurs a cost of 1 (using boolean notation which is 0 and 1). For each neighbor $j$ (i.e., $a_{ij} = 1$), there are three cases: 1) if $j$'s data is co-

located with $i$'s data, stay on the same $c$ to read $j$'s data; 2) if replica of $j$'s data is co-located with $i$'s data, stay on the same community $C$ to read $j$'s data; 3) neither of the above options, go to $j$'s community to read its data.

**Algorithm 1** Pseudocode of replica allocation

```
use GreedyApproach;
nodes are processed in the order of 1, 2, ..., N;
for all D_ic = 1 do
    g_i(c) = sum_{j=1}^{N} a_ij D_jc for i;
    Choose top X storage spaces of the community c
    if the storage space has highest g then
        Place a replica for user i;
    end if
    if there is a tie then
        Compute load l(c) = sum_{i=1}^{N} (D_ic + R_ic);
        Choose the storage with least load to place replica for user i.
    else
        Not enough X storage spaces in the histogram
    end if
end for
```

Therefore, based on the algorithm, a read query by user $i$ incurs the following cost while the write is always $X + 1$.

$$RC(i) = 1 + \sum_{j=1}^{N} a_{ij} \sum_{c=1}^{G} D_{ic}(1 - D_{jc})(1 - r_{jc})$$

Since the write cost is fixed for all nodes in the community, we measure the efficiency of our replica allocation method only in terms of the read cost for an average user. ComPAS, aims at providing the desired efficiency and consistency as the primary goal and to balance the community storage space load. Therefore, load for a storage space is proportional to the amount of data; original or replica, stored in the same space which is computed in the load balance computation and replication decision module, and in order to decide the most desirable location to place the replica of user $i$'s data, ComPAS computes the location histogram in its own module. In ComPAS, if we need to place a replica copy for a user $i$ somewhere, the most desirable location should be the primary storage place of most neighbors of $i$ (Algorithm 1). The algorithm works in a greedy manner for discovering the best way to place a replica by sequentially considering a node at a time. Greedy replica allocation is the simplest version, fast and robust approach, creating a number of replicas that look for the destination concurrently.

## 3. PRELIMINARY RESULTS

ComPAS has been compared to random replication scheme, W-DCG [3] and without replication. In Fig. 2(a), the read cost improves with replication. W-DCG's efficiency lowers when compared to ComPAS as the numbers of replicas are increasing. Therefore, ComPAS offers a more interesting pattern and it is the obvious superior scheme compared to the others. Data accessibility is the ratio of the number of successful access requests to the number of all access requests issued and it is an important criterion for a replication protocol. Fig. 2(b) shows that, as the number of nodes increases, the number of data items to be replicated also increases, hence the data accessibility reduces.

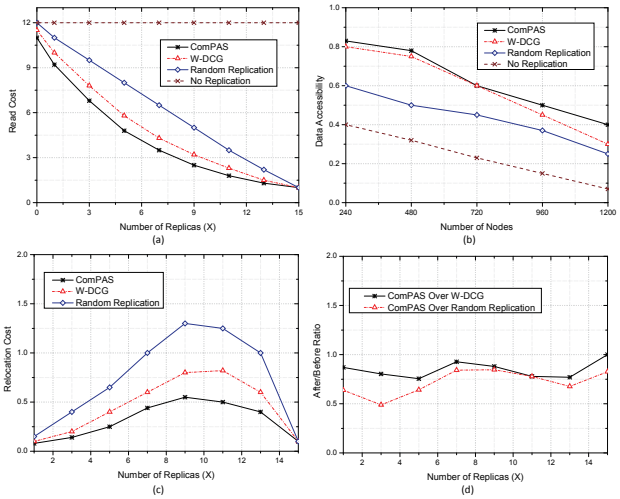


**Figure 2: Tests of (a) read cost, (b) data accessibility, (C) relocation cost, and (d) consistency.**

A desirable replica allocation method should keep the replica relocation cost as low as possible. The relocation cost is depicted in Fig. 2(c). It shows, there exists a special value of $X$ where the relocation cost is higher. This is because, when $X$ is closer to the two extreme values, there are too many replicas to relocate and not enough flexibility for where the replicas can be relocated. It is obvious that ComPAS is highly efficient when relocation happened. We have also performed an evaluation to measure the consistency of ComPAS's superiority over the others. We compute "after/before ratio" in Fig. 2(d). It implies, ComPAS is constantly superior to others. The results confirm the efficiency and consistency of the method that we have proposed. Finally, the poster presentation during the conference would also compare the results obtained for the variation of traffic with varying number of nodes and mobility groups which can be found in [1].

## 4. FUTURE WORK

In the future work, we aim to design a replication middleware with an optimal load balancing mechanism. We will also work on handling misbehaving nodes in the replica allocation operation. This would allow a reliable and more consistent service. ComPAS enables new features like cloud deployment, which we plan to implement in future as well.

## 5. REFERENCES


[1] F. Xia, A. M. Ahmed, L. T. Yang, J. Ma, and J. Rodrigues, Exploiting Social Relationship to Enable Efficient Replica Allocation in Ad-hoc Social Networks, *IEEE Transactions on Parallel and Distributed Systems*, 2014. DOI: 10.1109/TPDS.2013.2295805

[2] J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra and P. Rodriguez, The Little Engine(s) That Could: Scaling Online Social Networks, *Proc. of ACM SIGCOMM'10*, pp. 375-386, 2010.

[3] S. Jain, M. Chawla and N. Gupta, Weighted Dynamic Bi-connected Group based Replica Allocation method for Mobile Ad-hoc Networks, *Proc. of CAC2S*, pp. 189-197, 2013.