Modelling and Predicting the Data Availability in Decentralized Online Social Networks

Songling Fu[§], Ligang He[#*], Xiangke Liao[§], Chenlin Huang[§], Kenli Li[#], Cheng Chang[#], Bo Gao[*]

[§]School of Computer Science, National University of Defense Technology, Changsha, China
[*]Department of Computer Science, University of Warwick, Coventry, UK
[#]School of Information Science and Engineering, Hunan University, Changsha, China
e-mail: songling.fu@gmail.com, liganghe@gmail.com

Abstract—**Maintaining data availability is one of the biggest challenges in Decentralized Online Social Networks (DOSN). In the existing work of improving data availability in DOSN, it is often assumed that the friends of a user are always capable of contributing sufficient storage capacity to store all the data published by the user. However, this assumption is not always true for today's Online Social Networks (OSNs) for the following reasons. On one hand, the increasingly more data are being generated on the OSNs nowadays. On the other hand, current users often use the smart mobile devices to access the OSNs. These two factors cause the shortage of the storage capacity in DOSN, where the published data are supposed to be stored within a friend circle. The limitation of the storage capacity may jeopardize the data availability. Therefore, it is desired to know the relation between the storage capacity contributed by the OSN users and the level of data availability that the OSN can achieve. This paper addresses this issue. In this paper, the data availability model over storage capacity is established. Further, a novel method is proposed to predict the data availability on the fly. Extensive simulation experiments have been conducted to evaluate the effectiveness of the data availability model and the on-the-fly prediction. The data availability model can be used by the OSN designers to determine the storage capacity for the published data in order to achieve the desired data availability. The on-the-fly prediction method can help the data replication and storage policies make judicious decisions at runtime.**

Keywords— Decentralized Online Social Network; Data Availability; Prediction;

I. INTRODUCTION

In the last decade, Online Social Networks (OSNs), such as Facebook and Twitter, have gained extreme popularity with more than a billion users worldwide. OSNs allow a user to publish the data to all his friends in his friend circle.

Currently, the OSN platforms are typically centralized, where the users store their data in the centralized servers deployed by the OSN service providers. The service providers can utilize and analyze these data to know the users' private information, such as interest and personal affairs, and in the worst case may sell these information to the third party. Therefore, the current Centralized Online Social Networks (COSNs) have raised the serious concerns in privacy [11-14].

In order to address the data privacy issue, the Decentralized Online Social Networks (DOSNs) have been proposed recently [15,16]. Although the DOSN products [22] are not as popular as the OSNs, DOSN is indeed under active development [1, 17]. In order to protect the data privacy, the centralized servers are bypassed in DOSNs and the data published by a user are stored and disseminated only among the friend circle of the user [17]. Although DOSNs can help protect the data privacy, maintaining data availability becomes a big challenge. This is because if a friend of the user is offline, the data stored in the friend cannot be accessed by other friends.

In order to achieve good data availability in DOSN, the data replication approach has been widely used. In this approach, a certain number of data replicas are created for each data item published by a user and these data replicas are stored in the user's friend circle. By doing so, if a friend is offline, the data in this offline friend can be accessed through the replicated data stored in other friends. Consequently, data availability is improved.

In the existing data replication work in DOSN, it is typically assumed that the friends of a user are always capable of contributing sufficient storage capacity to store all the published data [17,18]. This assumption is not ideal in the current times. On one hand, the increasingly more data are being generated on the OSNs nowadays. On the other hand, the users now often use mobile devices, such as mobile phones, to access the OSN services. The storage capacity in the mobile devices is much more limited than the desktop computers used in the "old fashioned" style of accessing OSNs. Moreover, the number of the friends in a friend circle is limited [4]. These above factors cause the storage shortage in DOSNs. Therefore, it is desired to know what level of data availability can be achieved given the total storage capacity contributed by the friend circle. However, the existing work in DOSN has not yet conducted quantitative research in this aspect. This paper aims to address this issue and build a quantitative model to capture the relation between the total storage capacity contributed by the friends and the level of data availability in the DOSN.

Moreover, the friends become online or offline dynamically in a DOSN. The data availability will drop when the number of online friends decreases. A novel method is proposed in this paper to predict the level of data availability on the fly.

The reason why we investigate the relation between the total storage capacity and data availability is because a data item is regarded as being available as long as it is stored in DOSN, no matter which friends the data replicas are stored in. The location of the data replicas does not directly affect the data availability, but mainly imposes the impact in the following two aspects.

i) data accessing performance: Due to for example the bandwidth and latency of the friends where the data are

The corresponding author is Dr. Ligang He.

978-1-4799-5054-6/14 $31.00 © 2014 IEEE
DOI 10.1109/ICWS.2014.34

stored, other friends may who are accessing the data may experience different performance.

ii) the data maintenance overhead: When a friend goes offline, the data replicas on the friend have to be generated on other online friends. Various attributes of the friend, such as the storage capacity contributed by this friend, bandwidth and lantency, have impact.

How to optimize data accessing performance and reduce data maintenance overhead is the work of the underlying data replication and placement strategies. This work situates at the level of maintaining data availability. This is why in this work we mainly concern the total storage size provided by the friends collectively. Following on this work, we plan to work down the management levels in DOSNs and develop the placement strategies for data replicas among the friends.

Using the data availability model developed in this paper, the DOSN designers can determine the average size of the storage pool that each friend should contribute for the published data, given the level of data availability that the DOSN desires to achieve. Moreover, In DOSN, the friends become online and offline dynamically, the data availability will drop when the number of online friends decreases. The on-the-fly prediction method can be used to conduct the real-time prediction for the level of data availability in the near future. The quantitative prediction results produced by the model can greatly help the data replication and storage policies make judicious decisions on the fly.

The rest of this paper is organized as follows. Section II discusses related work about analyses of OSN properties, the existing DOSN approaches and data availability work. Section III states the problem which we try to address. Section IV presents the data availability model over storage capacity. Section V presents the on-the-fly prediction model. Section VI conducted extensive experiments to verify our models and analyzes experimental results. Finally, we make conclusions.

II. Related Work

A. Analyses of the OSN Properties

1) Characterizations of OSN networks

Some studies use the graphs to represent the OSN networks and investigate the graph structures of OSN, such as degree distribution, network diameter, clustering property and so on. They conduct the analyses through the crawled data gathered from popular OSN sites such as Facebook, Twitter, MySpace, Flickr, YouTube, LiveJournal, Cyworld and orkut [1-5]. It has been found that: i) OSNs manifest power-law, small-world properties; ii) The social network is nearly fully connected; iii) The neighborhoods of the users in the social graph contain the surprisingly dense structure, while the graph is sparse as a whole; iv) Most users have a moderate number of friends (less than 200). The findings about the number of friends will be used to design the simulation experiments in this paper.

2) Analyses of user behaviours

The work in [6-10] studied the patterns of the user behaviors through the crawled or clickstream data. Jin et al. [6] conducted a comprehensive review about the user behavior in OSNs from several perspectives, including social connectivity and interaction among users, traffic activity, and the characteristics in mobile environments. Benevenuto et al. [7] collected the clickstream data over 12 days to study the characteristics of OSN sessions, including the accessing frequency, session durations, and total time spent on OSNs. Schneider et al. [8] focused on feature popularity, session characteristics and the dynamics in the OSN sessions. Kwon et al. [9] empirically examined how the individual characteristics affect the actual user acceptance of social network services. Yan et al. [10] studied the human behavior in OSNs and found that the human activity patterns are heterogeneous and bursty, and often follow the power-law distribution.

Since the existing research has revealed the dynamic characteristics about user behaviors, such as the distributions of online and offline durations. These will be used as the known parameters when we derive the data availability model and the on-the-fly prediction in this paper.

B. DOSN

To address the data privacy problem in COSNs, several decentralized approaches have been proposed [15-16,19]. A distributed, peer-to-peer approach coupled with encryption is proposed in [15]. Reference [16] adopted a decentralized approach using the URIs as the identifiers throughout, which can provide the same (or even higher) level of user interaction as with many of the current popular OSN sties. None of these approaches only stores the data in his friend circle. Gemstone[19] stores the user's data in the so-called Data Holding Agents (DHAs). If a DHA itself is offline, the data have to be passed to the offline DHA's DHAs.

There are other types of DOSN [20], known as friend-to-friend storage systems, which focus on providing the data storage services for all participants. Li et al. [21] argued that a node should choose its neighbors where the data are stored based on existing social relationships instead of randomly. Our data availability model and the on-the-fly prediction can be integrated into these existing DOSNs, e.g., the quantitative results produced by our models can be used to help make the data replication and/or data storage decisions.

C. Data Availability in DOSN

Because of the requirement of protecting data privacy, the data published by a user are only stored in his friend circle in the DOSN. Consequently, data availability is one of the biggest challenges in DOSNs. The existing work in improving data availability mainly focuses on designing smart data replication and data storage policies.

The approach proposed by Koll et al. [17] exchanges the recommendations among the socially related nodes in order to effectively distribute a user's data replicas among the eligible nodes carefully selected in the OSN. In the approach developed by Olteanu et al. [18], the preferences are given to the nodes when it comes to selecting the nodes for storing the data (and their replicas) published by a user. Buchegger et al. designed a two-tiered DOSN architecture (PeerSoN) [15]. All the above existing work about data availability focuses on how to store the data replicas so that they are still accessible when the users or certain friends of the users are offline. They all implicitly assume that the friends are always able to contribute the adequate storage capacities to store the replicated data.

D. Data Availability in Grids and Clouds

We also studied the existing work in achieving data availability in Grids and Clouds. Amjad et al. [23] surveyed the dynamic replication strategies for improving data availability in data grids. CrossMann et al. [24] proposed a modular cloud storage system. However, the considerations in achieving data availability in Grids and Clouds are different from those in DOSNs. A big difference is that the data replication mechanisms in Grids or Clouds all explicitly or implicitly assume that the total storage space in Grids or Clouds is always sufficient to store the data replicas. This assumption is reasonable for Grids and Clouds because of the scale of such systems. However, it is not always true for DOSN due to 1) mobile devices are often used and 2) a friend circle is limited.

III. Problem Statement

Fig. 1 illustrates the data availability problem. In Fig. 1, the user publishes the data at a series of time points. Assume t_1 is the first time point when he publishes the data, $Data_1$, after he comes online, and t_k is the last time point the user publishes the data, $Data_k$, before he goes offline at the time point t_{out}^{u}. Now let's consider one of the friends in the user's friend circle. Assume that the friend goes offline at time point t_{out}^{f} just before the user publishes $Data_{k'}$ (and after the user publishes $Data_{k'-1}$), and then comes online at time point t_{in}^{f} after the user goes offline. Therefore, $Data_{k'}$ to $Data_k$ are the data that the friend missed when he is offline and consequently need to update when he comes online. Since the user is already offline, the friend can only update the missed data from other online friends where the data replicas are stored. Note that if the friend comes online before the user goes offline, the friend can update all missed data from the user directly. Therefore, data availability is not a problem under this circumstance.

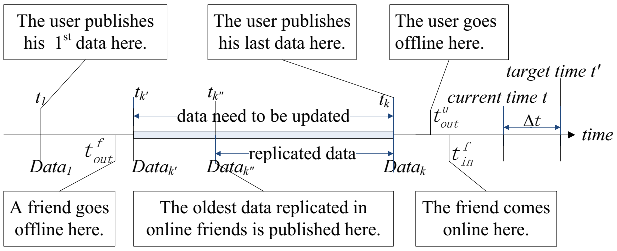

The user publishes his 1st data here.
The user publishes his last data here.
The user goes offline here.
target time t'
current time t
t_1
$t_{k'}$
$t_{k''}$
t_k
data need to be updated
t_{out}^{u}
Δt
time
t_{out}^{f}
replicated data
t_{in}^{f}
$Data_1$
$Data_{k'}$
$Data_{k''}$
$Data_k$
A friend goes offline here.
The oldest data replicated in online friends is published here.
The friend comes online here.

Fig. 1. The illustration of the data availability problem

When a friend comes online, assume that the total amount of the data that the friend tries to update is D_{update}. Out of D_{update}, the amount of data that are stored in online friends of the user is D_{stored}. The level of Data Availability (denoted by *DA*) is defined as Eq. (1).

$$DA = \frac{D_{stored}}{D_{update}} \tag{1}$$

The data replication frameworks typically work in the following way [17]. When the user publishes a data item, the certain number of data replicas are created and stored in the storage pools of the selected friends of the user. When a friend goes offline the data replicas which are stored in this friend will be recreated and stored on other online friends to maintain fixed number of data replicas for each data item. If the size of the storage pools is unlimited, the new data will just be added to the friend's storage pool. If the storage pool is limited and the pool is already full, the oldest data in the storage pool will be replaced with the new data. Therefore, the size of the storage pool will determine what period of data are stored in the pool, which affects the data availability of the DOSN. Consider Fig. 1 again, for example, if the storage pool in the friends is limited and can only store the data published from t_k back to $t_{k''}$, then the data earlier than $t_{k''}$ are not available when the friend comes online at t_{in}^{f}.

One aim of this paper is to establish the data availability model to capture the relation between the level of data availability and the total size of the storage pools contributed by the friends. This is presented in Section IV.

Now consider a time point t' after the current time t. Another aim of this paper is to predict the level of data availability at t' on the fly, which is presented in Section V. This prediction is very useful for the data replication or storage policies to make judicious decisions dynamically.

IV. The Data Availability Model Over Storage Capacity

As discussed in Section III, the total size of the storage pool contributed by a user's friends (denoted by *SS)* can determine the period of the published data stored in the storage pool. t_{tl} denotes the publishing time of the oldest data stored in the storage pool (i.e., $t_{k''}$ in Fig. 1), and t_{out}^{u} denotes the time when the user goes offline. Then $[t_{tl}, t_{out}^{u}]$ is the period of the published data stored in the storage pool. This section first determines t_{tl} (Section IV.A) and then presents the method of establishing the relation between *SS* and the *DA* of the data published by the user (Section IV.B).

A. Calculating t_{tl}

In order to determine t_{tl}, the size of the data published by the user has to be calculated first. $X(t_{pu})$ denotes the number of times that the user publishes the data in the time duration t_{pu}. $X(t_{pu})$ is a discrete random variable. $P_{pu}\left(x(t_{pu})\right)$ denotes the probability density function (pdf) of $X(t_{pu})$. a denotes the average size of the data published by the user each time. $S(t_{pu})$ denotes the total size of the data published by the user in t_{pu}. Clearly, $S(t_{pu}) = aX(t_{pu})$. Therefore, the pdf of $S(t_{pu})$, denoted by $S_{pu}\left(s(t_{pu})\right)$, can be determined by Eq. (2) and the expectation of $s(t_{pu})$ can be calculated by Eq. (3).

$$S_{pu}\left(s(t_{pu})\right) = a \cdot P_{pu}\left(x(t_{pu})\right) \tag{2}$$

$$\begin{aligned} E[S(t_{pu})] &= a \cdot E[X(t_{pu})] \\ &= a \cdot \sum_{x=1}^{+\infty} x \cdot P_{pu}(x(t_{pu})) \end{aligned} \tag{3}$$

The publishing time of the oldest data stored in the storage pool, t_{tl}, can be calculated using Eq. (4) given *SS*, where k is the replication degree in the OSN, i.e., the number of replicas created for each data item.

$$E[S(t_{out}^{u} - t_{tl})] \cdot k = SS \tag{4}$$

B. Establishing the relation between DA and SS

When a friend comes online at t_{in}^{f} (as in Fig. 1) and his last logout time (denoted by t_{out}^{f}) is no earlier than t_{tl}, the friend can update all the data missed during his offline duration from other online friends. Namely, *DA* for a friend coming online at t_{in}^{f}, denoted by $DA\left(t_{in}^{f}, t_{out}^{f}\right)$, is 100% in this case. When t_{out}^{f} is earlier than t_{tl}, the data published in $\left[t_{out}^{f}, t_{tl}\right]$ are not available to the friend. Therefore, *DA* in this case equals the proportion of the data that are published in $[t_{tl}, t_{out}^{u}]$ to those in $\left[t_{out}^{f}, t_{out}^{u}\right]$. In summary, $DA\left(t_{in}^{f}, t_{out}^{f}\right)$ can be calculated using Eq. (5).

$$DA\left(t_{in}^{f}, t_{out}^{f}\right) = \begin{cases} 100\% & t_{out}^{f} \geq t_{tl} \\ \dfrac{E[S(t_{out}^{u} - t_{tl})]}{E\left[S\left(t_{out}^{u} - t_{out}^{f}\right)\right]} \cdot 100\% & t_{out}^{f} < t_{tl} \end{cases} \quad (5)$$

t_{off} denotes the time duration of a friend being offline continuously. $f_{off}\left(t_{off}\right)$ denotes the pdf of t_{off} . The probability that a friend went offline at t_{out}^{f} and then comes online at t_{in}^{f} is $f_{off}\left(t_{in}^{f} - t_{out}^{f}\right) dt_{out}^{f}$ and the corresponding $DA\left(t_{in}^{f}, t_{out}^{f}\right)$ is obtained by Eq. (5). Then, *DA* at time point t_{in}^{f} can be expressed by Eq. (6).

$$\int_{t_{out}^{u}}^{0} f_{off}\left(t_{in}^{f} - t_{out}^{f}\right) \cdot DA(t_{in}^{f}, t_{out}^{f}) dt_{out}^{f} \quad (6)$$

$DA_{[t_{out}^{u},h]}$ denotes the expectation of *DA* over the time duration between t_{out}^{u} and t_{in}^{f}, where *h* is the duration between the user's two consecutive logins (The work in [8] has presented the method to obtain the value of *h*). $DA_{[t_{out}^{u},h]}$ can be calculated by Eq. (7), where $f_{at}\left(t_{in}^{f}\right)$ is the probability density function that a friend comes online at time t_{in}^{f}.

$$DA_{[t_{out}^{u},h]} = \int_{t_{out}^{u}}^{h} f_{at}(t_{in}^{f}) \cdot \int_{t_{out}^{u}}^{0} f_{off}\left(t_{in}^{f} - t_{out}^{f}\right) DA(t_{in}^{f}, t_{out}^{f}) dt_{out}^{f}\, dt_{in}^{f} \quad (7)$$

$DA_{[0,t_{out}^{u}]}$ denotes the expectation of *DA* over the time duration between 0 and t_{out}^{u}. Since the user is online between 0 and t_{out}^{u}, *DA* is 100% over the time duration between 0 and t_{out}^{u}, i.e., Eq. (8) holds.

$$DA_{[0,t_{out}^{u}]} = 100\% \quad (8)$$

t_{on} denotes the time duration of a friend being online continuously. $f_{on}(t_{on})$ denotes the pdf of t_{on}. *DA* of the data published by the user under the given value of *h*, denoted by $DA(h)$, can be calculated by combining Eq. (7) and (8) as follows.

$$DA(h) = \int_{0}^{h} f_{on}(t_{out}^{u}) \cdot \left(\frac{t_{out}^{u}}{h} \cdot DA_{[0,t_{out}^{u}]} + \frac{h - t_{out}^{u}}{h} \cdot DA_{[t_{out}^{u},h]}\right) dt_{out}^{u} \quad (9)$$

$h = t_{on} + t_{off}$ is also a random variable. $f_{H}(h)$ denotes the probability density function of *h*, which can be derived from the probability density functions of t_{on} and t_{off} and has also been studied in the literature [9].

Therefore, *DA* of the data published by the user can be finally calculated using Eq. (10).

$$DA = \int_{0}^{\infty} DA(h) \cdot f_{H}(h) dh \quad (10)$$

As can be seen from Eq. (9), *DA* is a function over $DA_{[t_{out}^{u},H]}$, which is in turn a function over $DA\left(t_{in}^{f}, t_{out}^{f}\right)$ (shown in Eq. (7)). $DA\left(t_{in}^{f}, t_{out}^{f}\right)$ is the function over t_{tl} (Eq. (5)). As shown in Eq. (4), t_{tl} can be calculated from *SS*. Thus, we have now established the function of *DA* over *SS*.

V. PREDICTING THE DATA AVAILABILITY ON THE FLY

Using the method in Section IV, we can calculate *SS* required to achieve the desired *DA* of the data published by the user. Note that *SS* is the total storage capacity of all online friends of the user. The friends log in and out dynamically and therefore the number of online friends varies over time. When the number of online friends decreases, the size of the individual storage pool contributed by each online friend has to be increased in order to maintain the desired *DA*. The existing work in the literature often assumes that the friends of a user are always capable of contributing sufficient storage capacity for the replicated data published by the user. Consequently, there is little work yet in the literature investigating the impact of the friends' dynamic behaviors (i.e., dynamic login and logout) on *DA*. However, as we have discussed in the introduction section, it is not always acceptable to assume that the friends are willing and able to contribute unlimited storage capacity in the nowadays OSNs. In this paper, we assume that the maximum storage capacity that each friend is able to contribute is *S*. When the required *SS* exceeds the total storage capacity contributed by all online friends, the *DA* will drop. Due to the friends' dynamic behaviors, it is very useful to be able to predict the *DA* on the fly. This section addresses this issue. Consider Fig. 1 again. Assume the current time is t. The problem of the on-the-fly prediction of *DA* is to predict the *DA* at a future time point t' $(t' > t)$.

According to the discussions above, the key of predicting *DA* is to predict the number of online friends. At the current time t, we know how many friends are online or offline. We can predict the number of friends who are online at a future time t', if we can predict the following two parameters: i) how many of the friends who are online at time t do not change their states from online to offline before or at t', and ii) how many of the friends who are offline at time t change their states to online before or at t'. The methods of predicting the above two parameters are presented in Section V.A and V.B, respectively. Section V.C combines the results obtained in Section V.A and V.B to predict the number of online friends and further predict the *DA* at time t'.

A. Predicting the number of the friends who are online at t *and do not change to offline before or at* t'

Given an online friend v_i at time t, we can know the time point at which the friend logged in (i.e., became online), which is denoted by $t_{in\_i}^{on}$. The probability that friend v_i does not change to offline before t' equals the probability that v_i will only log out after t' (i.e., v_i's logout time, denoted by $t_{out\_i}^{on}$ is greater than t'). The probability, denoted by $p_{out\_i}^{on}\left(t_{out\_i}^{on} > t'\right)$, in turn equals the probability that v_i's online duration is greater than $\left(t' - t_{in\_i}^{on}\right)$ under the condition that v_i's online duration is no less than $\left(t - t_{in\_i}^{on}\right)$, which can be computed using the conditional probability shown in Eq. (11). The condition of $\left(t_{on} \geq t - t_{in\_i}^{on}\right)$ in Eq. (11) reflect the fact that v_i has been staying online for the duration of $\left(t - t_{in\_i}^{on}\right)$.

$$\begin{aligned} p^{on}_{out\_i}\left(t^{on}_{out\_i} > t'\right) &= p_{on}\left(\left(t_{on} > t' - t^{on}_{in\_i}\right)\middle|\left(t_{on} \ge t - t^{on}_{in\_i}\right)\right) \\ &= \frac{1 - F_{on}\left(t' - t^{on}_{in\_i}\right)}{1 - F_{on}\left(t - t^{on}_{in\_i}\right)} \qquad (11) \end{aligned}$$

V_{on} and N_{on} denotes the set and the number of all online friends at time t, respectively. Then the number of the friends in V_{on} who are still online at t' can be predicted using Eq. (12).

$$\sum_{i=1}^{N_{on}} p^{on}_{out\_i}\left(t^{on}_{out\_i} > t'\right) \qquad (12)$$

B. Predicting the number of the friends who are offline at t and change the states to online before or at t′

The method of predicting the number of the friends who are offline at t and change the states to online before or at t' is similar as that presented in Section V.A.

$$\begin{aligned} p^{off}_{in\_j}\left(t^{off}_{in\_j} \le t'\right) &= p_{off}\left(\left(t_{off} \le t' - t^{off}_{out\_j}\right)\middle|\left(t_{off} \ge t - t^{off}_{out\_j}\right)\right) \\ &= \frac{F_{off}\left(t' - t^{off}_{out\_j}\right) - F_{off}\left(t - t^{off}_{out\_j}\right)}{1 - F_{off}\left(t - t^{off}_{out\_j}\right)} \qquad (13) \end{aligned}$$

Given an offline friend v_j at time t, we can know the time when v_j logged off, denoted by $t^{off}_{out\_j}$. The probability that v_j changes the state to online before or at t' equals the probability that v_j's login time, $t^{off}_{in\_j}$, is no later than t'. The probability, denoted by $p^{off}_{in\_j}\left(t^{off}_{in\_j} \le t'\right)$, in turn equals the probability that v_j's offline duration is smaller than $\left(t' - t^{off}_{out\_j}\right)$ under the condition that v_j's offline duration is no less than $\left(t - t^{off}_{out\_j}\right)$, which can be calculated using Eq. (13).

V_{off} and N_{off} denotes the set and the number of all offline friends at time t, respectively. Then the number of the friends in V_{off} who change the states to online before or at time t' can be predicted using Eq. (14).

$$\sum_{j=1}^{N_{off}} p^{off}_{in\_j}\left(t^{off}_{in\_j} \le t'\right) \qquad (14)$$

C. Predicting the number of online friends and the DA at t′

$N_{on}(t')$ denotes the number of online friends at t'. $N_{on}(t')$ can be calculated by Eq. 15 by combining (12) and (14).

$$\begin{aligned} N_{on}(t') &= \sum_{i=1}^{N_{on}} p^{on}_{out\_i}\left(t^{on}_{out\_i} > t'\right) + \sum_{j=1}^{N_{off}} p^{off}_{in\_j}\left(t^{off}_{in\_j} \le t'\right) \\ &= \sum_{i=1}^{N_{on}} \left(\frac{1 - F_{on}\left(t' - t^{on}_{in\_i}\right)}{1 - F_{on}\left(t - t^{on}_{in\_i}\right)}\right) \\ &\quad + \sum_{j=1}^{N_{off}} \left(\frac{F_{off}\left(t' - t^{off}_{out\_j}\right) - F_{off}\left(t - t^{off}_{out\_j}\right)}{1 - F_{off}\left(t - t^{off}_{out\_j}\right)}\right) \qquad (15) \end{aligned}$$

S is the maximum storage capacity that each friend is able to contribute. Then the total storage capacity contributed by all online friends at time t' is $\left(S \cdot N_{on}(t')\right)$. Using the method presented in Section IV, the *DA* at t' can be determined.

VI. Case Study

When we derive the *DA* model over storage capacity and the on-the-fly prediction of *DA* in Section IV and V, we used the generic form of the probability distribution for online and offline durations (i.e., $f_{on}(t_{on})$ and $f_{off}(t_{off})$) as well as for the data publishing pattern (i.e., $X(t_{pu})$. However, it has been shown that the online and offline durations may follow the power-law distribution or the exponential distribution [25, 26], and that the data publishing pattern may follow the Poisson process [26]. In this section, we conduct the case studies by substituting the generic form of the probability distribution for the power-law and the Poisson distribution. In fact, any probability distributions can be used in the constructed models. Even if the mathematical derivations may not be conducted with some distributions, *Mathematica*[27] can be used to calculate the model results numerically.

A. Poisson distribution

If the data publishing pattern, $X(t_{pu})$, follows the Poisson distribution with the parameter λ_{pu}, then we have Eq. (16). Consequently, $\mathrm{E}\left[X(t_{pu})\right]$ can be calculated using Eq. (17).

$$P_{pu}\left(x(t_{pu})\right) = e^{-\lambda_{pu} t_{pu}} \frac{\left(\lambda_{pu} t_{pu}\right)^x}{x!} \qquad (16)$$

$$E\left[X(t_{pu})\right] = \lambda_{pu} t_{pu} \qquad (17)$$

Further, Eq. (3) can be transformed to Eq. (18).

$$E\left[S(t_{pu})\right] = a \cdot E\left[X(t_{pu})\right] = a\lambda_{pu} t_{pu} \qquad (18)$$

With Eq. (18), Eq. (4) becomes Eq. (19).

$$ak\lambda_{pu}\left(t^u_{out} - t_{tl}\right) = SS \qquad (19)$$

Therefore, t_{tl} can be calculated using Eq. (20).

$$t_{tl} = t^u_{out} - \frac{SS}{ak\lambda_{pu}} \qquad (20)$$

B. power-law distribution

If the offline duration, t_{off} , follows power-law with parameter λ_{off} , then we have Eq. (22), where $\mathrm{c} = \left(\lambda_{off} - 1\right) {t_{min}}^{\lambda_{off} - 1}$ given the minimal duration t_{min} [25].

$$f_{off}\left(t_{off}\right) = c \cdot {t_{off}}^{-\lambda_{off}} \qquad (22)$$

We now show how to use the power-law distribution to derive the on-the-fly prediction for the number of online friends, which is obtained using Eq. (11), (13) and (15).

Eq. (11) can be further derived to obtain Eq. (23).

$$\begin{aligned} p^{on}_{out\_i}\left(t^{on}_{out\_i} > t'\right)_{pl} &= \frac{1 - \int_{t_{min}}^{t' - t^{on}_{in\_i}} c {t_{on}}^{-\lambda_{on}} dt_{on}}{1 - \int_{t_{min}}^{t - t^{on}_{in\_i}} c {t_{on}}^{-\lambda_{on}} dt_{on}} \\ &= \left(\frac{t' - t^{on}_{in\_i}}{t - t^{on}_{in\_i}}\right)^{1-\lambda_{on}} \qquad (23) \end{aligned}$$

Eq. (13) can be further derived to obtain Eq. (24).

$$\begin{aligned} p^{off}_{in\_j}\left(t^{off}_{in\_j} \le t'\right)_{pl} &= \frac{\int_{t - t^{off}_{out\_j}}^{t' - t^{off}_{out\_j}} c {t_{off}}^{-\lambda_{off}} dt_{off}}{1 - \int_{t_{min}}^{t - t^{off}_{out\_j}} c {t_{off}}^{-\lambda_{off}} dt_{off}} \\ &= 1 - \left(\frac{t' - t^{off}_{out\_j}}{t - t^{off}_{out\_j}}\right)^{1-\lambda_{off}} \qquad (24) \end{aligned}$$

Eq. (15) can be further derived to Eq. (25).

$$N_{on}(t')_{pl} = \sum_{i=1}^{N_{on}} \left(\frac{t' - t_{in\_i}^{on}}{t - t_{in\_i}^{on}}\right)^{1-\lambda_{on}} + \sum_{j=1}^{N_{off}} \left(1 - \left(\frac{t' - t_{out\_j}^{off}}{t - t_{out\_j}^{off}}\right)^{1-\lambda_{off}}\right) \quad (25)$$

VII. Evaluation

A discrete simulator has been developed in this work to simulate a DOSN. There are *N* users in the simulated DOSN. Some users act as the friends of another user and update the data published by the user. The online and offline durations of the users in the simulated DOSN follow the Power-Law distribution (PL) or the Exponential distribution (Exp), as observed in the literature. The user publishes the data following the Poisson process and *k* copies of replicas are created for each data item and stored in the online friends.

In order to evaluate the *DA* model over storage capacity, the *DA* is predicted given the size of storage capacity and other parameters values. Then the simulated DOSN is run using those parameters values. Each friend offers the same storage capacity, which can be adjusted so that the total storage capacity of all online friends always equals the storage capacity used to predict the *DA*. During the running, when a friend comes online at a time point, the *DA* of the published data for the friend is recorded. The average of all recorded *DA* is regarded as the actual *DA*, which is compared against the predicted *DA* to measure the prediction accuracy.

In order to evaluate the on-the-fly prediction, the experimental scenario is designed as follows. A user and his friends log in and out following the specified distribution during the time interval $[0, l]$. The current time is set to be *m*-th min ($m < l$ and the user is offline at time *m*). The online or offline states of all friends at time *m* as well as the latest login or logout time before time m are collected. The collected data, combing with the specified distributions of the friends' online and offline duration, are used to predict the number of online friends and *DA* at the future time points (i.e., the time points later than *m*). The predicted data are then compared against the data obtained from the actual running. For example, the number of the friends of a user is set to be 150. Fig. 2 shows the online/offline state of each friend when the current time is set to be 31st min. A point above the red line (i.e., when y=0) represents the latest login time of a friend who is online at 31st min, while a point below the red line shows the latest logout time of a friend who is offline at 31st min.

In the rest of this section, the *DA* model over storage capacity is evaluated in Section VII.A with regards to the following aspects: i) the impact of storage capacity on *DA*, ii) the impact of the DOSN parameters, including online /offline duration and the rate of user publishing data, on *DA*, and iii) the accuracy of the relation established between *DA* and *SS*.

In Section VII.B, the on-the-fly prediction is evaluated with regards to the following aspects: i) the accuracy of predicting the number of online friends on the fly, ii) the accuracy of the *DA* predicted on the fly.

Unless stated otherwise, the experimental parameters used in the evaluations take the values shown in Table II. These values are chosen based on those used in the literature [4].

TABLE I. Default values of the experimental parameters

Notations	Value	Descriptions
N	150	The number of the user's friends.
a	1	The average size of published data
λ_{on}^{exp}	1/3	The parameter of the online time duration which follows exponential distribution
λ_{off}^{exp}	1/11	The parameter of the offline time duration which follows exponential distribution
λ_{on}^{pl}	2.5	The parameter of the online time duration which follows power-law distribution
λ_{off}^{pl}	2.1	The parameter of the offline time duration which follows power-law distribution
λ_{pu}^{ps}	1	The parameter of the number of times the user publishes data which follows Poisson distribution

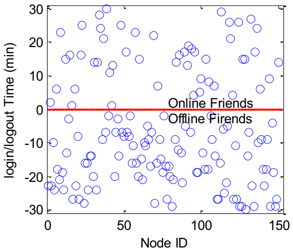


Fig. 2. The states of all friends at current time point

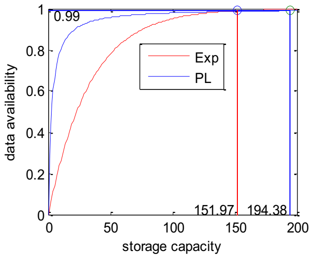


Fig. 3. The impact of *SS* on *DA*

A. Evaluating the DA model over storage capacity

1) Impact of storage capacity on DA

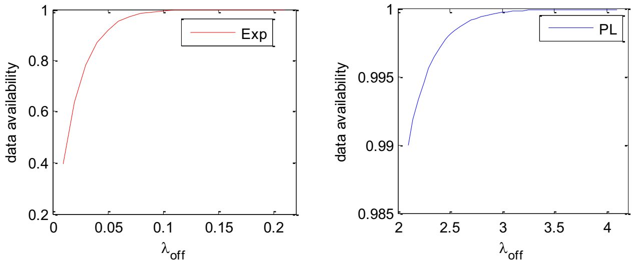
Fig. 3 shows the impact of the total storage capacity (i.e., *SS* in Section IV) on the *DA* calculated from the *DA* model presented in Section IV. As shown in Fig. 3, the *DA* increases as *SS* increases. Under both Exponential distribution and Power-Law distribution of the friends' online duration, data availability tails off after *SS* increases more than a certain value. These results suggest that it is unnecessary to ask the friends to contribute unlimited storage capacity, as often assumed in the work in the literature [16,17].

From this figure, we can also determine *SS* that is required to achieve a certain *DA*. For example, *DA* reaches 99% under PL or Exp when *SS* is 194.38 and 151.97, respectively.

2) Impact of on/offline durations on DA

As can be seen from the derivation of the *DA* model in Section IV, the online/offline durations impact on *DA*. We conducted the experiments to evaluate their impact. Since the online and offline durations have the similar impact, only the results for offline durations are presented here. Given the distribution, the average duration is controlled by λ_{off}. The inverse of λ_{off} is the length of the duration.

Fig. 4 shows the impact of λ_{off} on *DA*. In the experiments in Fig. 4, *SS* is set to be 194.38 and 151.97 under PL and Exp (as shown in Fig. 3), respectively, so that *DA* is 99% under the default value of λ_{off} (as in Table II). We then change the value of λ_{off} and plot the corresponding *DA*. It can be seen that *DA* increases as λ_{off} increases under both Exp and PL. These results can be explained as follows. When λ_{off} increases, the average length of the friends' offline durations decreases. Given a certain *SS*, the period of the stored data (i.e., $[t_{tl}, t]$) is fixed. Thus, the shorter offline durations of the friends result in higher probability that the time of the data that the friends try to update fall into $[t_{tl}, t]$. Consequently, *DA* is higher.



(a) exponential distribution (b) power-law distribution

Fig. 4. The impact of the offline durations on *DA*

3) *Impact of the data publishing rate on DA*

From the *DA* model, we can also know that the pattern with which the user publishes data has the impact on *DA*. It is shown in the literature that the number of times that the user publishes the data in a duration follows the Poisson distribution. Then, the parameter of the Poisson distribution, λ_{pu}, reflects the data publishing rate. Higher λ_{pu} means a higher rate.

Fig. 5 demonstrates the impact of λ_{pu} on *DA*. The setting of *SS* is the same as that in Fig. 4. The figure shows that *DA* decreases as λ_{pu} increases. This is because when the data are published at a higher rate, $[t_{tl}, t]$ is shorter given a fixed *SS*. Consequently, *DA* is lower.

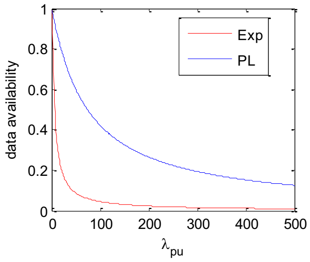
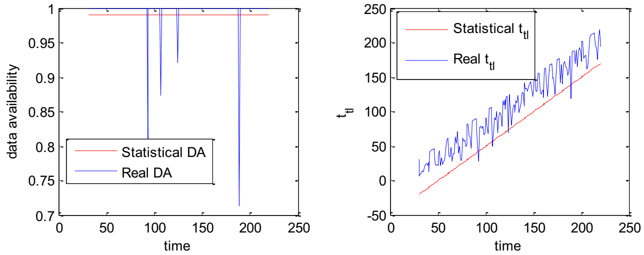


Fig. 5. The impact of the data publishing rate on *DA*

4) *Accuracy of the DA model*

The *DA* model over storage capacity in Section IV can calculate the *DA* given a *SS*. We conducted the experiments to study how accurate the calculated *DA* is, compared with the *DA* obtained from the actual running. The results are presented in Fig. 6. The results under Exp and PL show the similar pattern. Therefore, only the results under Exp are presented.

In Fig. 6, the setting of *SS* is the same as that in Fig. 4 (i.e., 151.97). The *DA* calculated by the *DA* model is 99%, which is the red line in Fig.6a. We run the simulated OSN with this *SS* and plot the actual *DA* over time, which is the blue line in Fig. 6a. It can be seen that the *DA* is fairly close to the calculated *DA* in most cases. These results suggest that the *DA* model is effective. In order to reveal the fundamental reason for this, we also compared t_{tl} obtained in the *DA* model (the red line in Fig. 6b) with the time of the oldest data that a friend tried to update when he came online at a time point (plotted in blue in Fig. 6b). If the time of the oldest data is not earlier than the calculated t_{tl}, the *DA* model is effective. As can be seen from Fig. 6b, the blue line are higher (i.e., the corresponding time is later) than the red line in most cases. This gives the fundamental reason why the *DA* model is effective, i.e., with the *SS* obtained by the *DA* model, the online friends can in most cases store the data that a friend tries to update when he comes online.



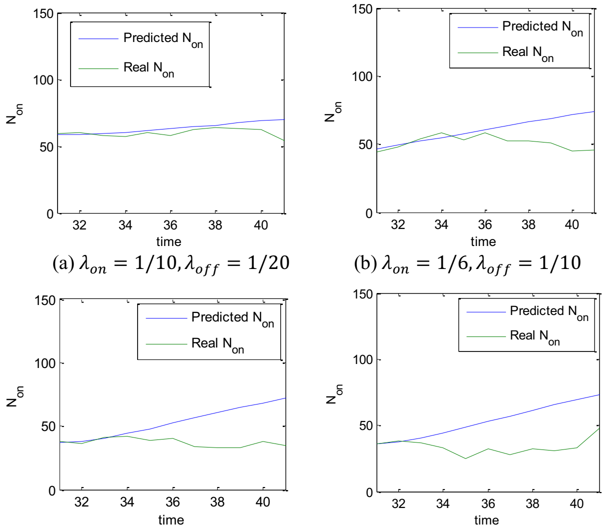
(a) From the perspective of *DA* (b) From the perspective of t_{tl}

Fig. 6. The Accuracy of the *DA* model

B. *Evaluating the on-the-fly prediction of DA*

1) *Accuracy of the predicted number of online friends and the impact of online and offline durations*

As shown in Section V, the predicted number of online friends (i.e., N_{on}) determines the value of the on-the-fly *DA*. Therefore, we conducted the experiments to evaluate the accuracy of predicting N_{on}. The experimental scenario has been presented in the third paragraph of Section VII. The experimental results are shown in Fig. 7.



(a) $\lambda_{on} = 1/10, \lambda_{off} = 1/20$ (b) $\lambda_{on} = 1/6, \lambda_{off} = 1/10$

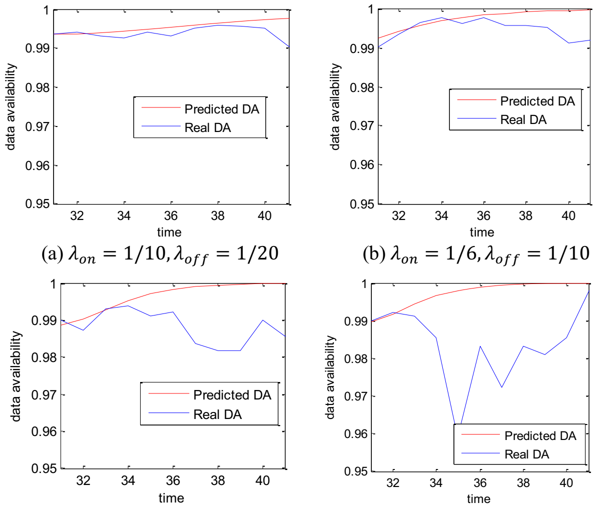
(c) $\lambda_{on} = 1/3, \lambda_{off} = 1/11$ (d) $\lambda_{on} = 1, \lambda_{off} = 1/15$

Fig. 7. The accuracy of prediction model over time

In Fig. 7, the current time point is set to be 31st min and the on-the-fly prediction predicts N_{on} from 31st min onwards, which is plotted in blue. The actual N_{on} from 31st min onwards is plotted in green. Fig. 7a, b, c and d show the results under different λ_{on} and λ_{off} (i.e., online and offline durations). It can be seen from Fig. 7a that compared with its actual values, the prediction of N_{on} is fairly accurate in the first 10 minutes, which shows the effectiveness and applicability of the proposed prediction method since the prediction can be conducted on the fly as the time elapses. By comparing Fig. 7a, b, c and d, we can see that the length of the accurate prediction decreases as the settings of λ_{on} and λ_{off} change from Fig. 7a to 7d. These results indicate that the online and offline durations have impact on the prediction accuracy. After carefully analyzing the changing trend of λ_{on} and λ_{off}, it appears the minimum value between the online and the offline durations (i.e., $\min(1/\lambda_{on}, 1/\lambda_{off})$) determines the length of accurate prediction. The less value of $\min(1/\lambda_{on}, 1/\lambda_{off})$, the shorter length of the accurate prediction. The reason for this is because when $\min(1/\lambda_{on}, 1/\lambda_{off})$ is smaller, the friends are more dynamic and consequently, it is more difficult to obtain the accurate prediction in the future.

2) *Accuracy of the predicted DA*

Finally, Fig. 8 presents the experiments results that show the accuracy of the on-the-fly prediction of *DA*. The experimental settings in Fig. 8 are the same as those in Fig. 7. It can be seen from Fig. 8, the trends shown in Fig. 8 are consistent with those in Fig. 7. This once again shows the effectiveness of the on-the-fly prediction.



(a) $\lambda_{on} = 1/10, \lambda_{off} = 1/20$ (b) $\lambda_{on} = 1/6, \lambda_{off} = 1/10$

(c) $\lambda_{on} = 1/3, \lambda_{off} = 1/11$ (d) $\lambda_{on} = 1, \lambda_{off} = 1/15$

Fig. 8. The accuracy of the on-the-fly prediction of *DA*

VIII. Conclusions

This paper proposes a data availability model over storage capacity for DOSNs. Further, a novel method is proposed to predict the data availability on the fly. Extensive simulation experiments have been conducted. The results show that the proposed data availability method is able to capture the relation between data availability and storage capacity effectively, and that the on-the-fly prediction method can predict the level of data availability accurately.

This work is situated at the level of maintaining the data availability. How to optimize the data accessing performance and reduce the data maintenance overhead is the work of the underlying data replication and placement strategies. In the future, we plan to work down the management level in DOSN and develop the strategies of placing data replicas among friends in DOSN. When designing the placement strategies, the attributes of individual friends, such as the bandwidth and latency associated to a friend, the storage capacity contributed by a friend and so on, will be taken into account.

Acknowledgment

We would like to thank the users and the developer community for their help with this work. The work reported in this paper was supported by China HGJ Project (No. 2013ZX01040-002), China Open Fund Project (No. KJ-13-105), and China NSFC No.61304119, No.61303191, No. 61272482, No. 61272483 and No.61379146.

References

[1] R. E. Wilson, S. D. gosling, L. T. Graham, A Review of Facebook Research in the Social Sciences, Perspectives on Psychological Science, 7(3): 203-220, May 2012.

[2] McGlohon M, Akoglu L, Faloutsos C. Statistical properties of social networks[M]//Social Network Data Analytics. Springer US, 2011: 17-42.

[3] Ahn Y Y, Han S, Kwak H, et al. Analysis of topological characteristics of huge online social networking services, ACM WWW'07, 835-844.

[4] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the Facebook social graph. CoRR, abs/1111.4503, 2011.

[5] Mislove A, Marcon M, Gummadi K P, et al. Measurement and analysis of online social networks, ACM SIGCOMM'07, 2007: 29-42.

[6] Jin L, Chen Y, et al. Understanding user behavior in online social networks: A survey. IEEE Communications Magazine, 2013: 144-150.

[7] Benevenuto F, Rodrigues T, Cha M, et al. Characterizing user behavior in online social networks. ACM SIGCOMM'09, 2009: 49-62.

[8] Schneider F, Feldmann A, et al. Understanding online social network usage from a network perspective. ACM SIGCOMM'09, 2009: 35-48.

[9] Kwon O, et al. An empirical study of the factors affecting social network service use. Computers in Human Behavior, 2010, 26(2): 254-263.

[10] Yan, Q., Wu, L., Liu, C., Li, X, Information Propagation in Online Social Network Based on Human Dynamics. In *Abstract and Applied Analysis* (Vol. 2013). Hindawi Publishing Corporation, May 2013.

[11] Krishnamurthy B. Privacy and Online Social Networks: Can colorless green ideas sleep furiously? IEEE Security &Privacy, 2013, 11(3): 14-20.

[12] Krishnamurthy B, Wills C E. Characterizing privacy in online social networks. ACM WOSN, 2008: 37-42.

[13] Krishnamurthy B, Wills C E. Privacy leakage in mobile online social networks, The 3rd Conf. on Online social networks. USENIX Association, 2010.

[14] Zhang C, Sun J, et al. Privacy and security for online social networks: challenges and opportunities. Network, IEEE, 2010, 24(4): 13-18.

[15] Buchegger S, Schiöberg D, Vu L H, et al. PeerSoN: P2P social networking: early experiences and insights. ACM WOSN'09: 46-52.

[16] Yeung C A, Liccardi I, Lu K, et al. Decentralization: The future of online social networking. W3C Workshop on the Future of Social Networking, 2009.

[17] David Koll, Jun Li, Xiaoming Fu, With a Little Help From my Friends: Replica Placement in Decentralized Online Social Networks, TR-IFI-TB-2013-01, University of Goettingen, Germany, January 2013.

[18] Olteanu A, Pierre G. Towards robust and scalable peer-to-peer social networks. ACM WOSN, 2012.

[19] Tegeler F, Koll D, Fu X. Gemstone: Empowering Decentralized Social Networking with High Data Availability, IEEE GLOBECOM 2011: 1-6.

[20] Li J, Dabek F. F2F: Reliable Storage in Open Networks. IPTPS. 2006.

[21] Sharma R, Datta A, DeH'Amico M, et al. An empirical study of availability in friend-to-friend storage systems. IEEE P2P 2011: 348-351.

[22] Diaspora, https://joindiaspora.com/

[23] Amjad T, Sher M, Daud A. A survey of dynamic replication strategies for improving data availability in data grids. Future Generation Computer Systems, 2012, 28(2): 337-349.

[24] Kossmann D, Kraska T, Loesing S, et al. Cloudy: A modular cloud storage system[J]. Proc. of the VLDB Endowment, 2010.

[25] Zhou T, Han X P, Wang B H. Towards the understanding of human dynamics[J]. Science matters: humanities as complex systems, 2008: 207-233.

[26] Stutzbach D, Rejaie R. Understanding churn in peer-to-peer networks. Proceedings of the 6th ACM SIGCOMM conference on Internet measurement. ACM, 2006: 189-202

[27] *Mathematica* software, http://www.wolfram.com/