Privacy Preserving Efficient Data Dissemination in Decentralized Online Social Networks

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering

by

Esra Erdin

Dr. Mehmet Hadi Gunes / Dissertation Advisor

August, 2017
We recommend that the dissertation prepared under our supervision by

ESRA ERDIN

Entitled

Privacy Preserving Efficient Data Disseminaion
In Decentralized Online Social Networks

be accepted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Mehmet Hadi Gunes, Ph.D., Advisor

Murat Yuksel, Ph.D., Committee Member

Monica Nicolescu, Ph.D., Committee Member

Ming Li, Ph.D., Committee Member

Yaakov Varol, Ph.D., Committee Member

M. Sami Fadali, Ph.D., Graduate School Representative

David W. Zeh, Ph. D., Dean, Graduate School

August, 2017
Abstract

A growing concern for users of Online Social Networks (OSN) is the privacy and control of user data due to the client-server architecture of the popular ecosystems. In this dissertation, we discuss how to develop a decentralized OSN platform so that the user has direct control of his/her information. When designing a new OSN, it is critical to understand how users interact over the OSN platform and provide the users with similar functionality. To this end, we analyzed online activities of Facebook volunteers to obtain a better understanding of the interactions in the user’s friend circle. We performed a feasibility study to assess whether storage clouds can provide speeds comparable to popular OSNs, and observed that storage clouds have better performance than OSNs with delivering larger data while has comparable performance with smaller objects.

We then, propose a privacy preserving decentralized Personal Online Social Network (POSN). POSN removes the central authorities from the OSN as users share their content with intended peers through smart devices and cloud backend. POSN takes advantage of free storage clouds to efficiently distribute encrypted user data and provide timely access to the content. The proposed scheme ensures that data is accessible only to the desired friends and gives the user full control over their data access. Access control is directly managed by the user via the cloud API and removes centralized identity management that exists even in some of the decentralized architectures. Data processing is handled by smartphones, since they not only provide continuous Internet access but also possess computational power reaching similar
levels of traditional computers.

We developed access control mechanisms where user’s identity is ensured through their email address or phone number and fine-grained access provided through a hierarchy of access tokens. We also analyzed content dissemination in decentralized OSNs and developed mechanisms that rely on friendship communities to aggregate information from and reduce overhead on users. Finally, we developed a low latency data dissemination mechanism through online users and friendship communities so that smaller objects such as comments are disseminated in a timely manner.
Acknowledgments

I would like to sincerely thank my advisor Dr. Mehmet Hadi Gunes for his advising, endless support and patience from the day I came to Reno, throughout completion of my degree. I would like to thank my dissertation committee of Dr. Murat Yuksel, Dr. Monica Nicolescu, Dr. Yaakov Varol, Dr. Ming Li and Dr. Sami Fadali for their feedback and suggestions. I am grateful to my family for all the love, support and encouragement.

Esra Erdin

University of Nevada, Reno
August 2017
## Contents

Abstract

Acknowledgments

List of Tables

List of Figures

Chapter 1 Introduction

Chapter 2 Decentralized Online Social Networks

2.1 Overview on Social Networks

2.2 Decentralized Online Social Networks

2.2.1 Cachet

2.2.2 Persona

2.2.3 Safebook

2.2.4 Peerson

2.2.5 Diaspora

2.2.6 My3

2.2.7 Tent
List of Tables

2.1 Comparison of decentralized OSNs ................................................. 18
3.1 Posts by the user and user's circles .............................................. 22
5.1 Number of posts and post size for one friend .................................. 49
7.1 Average Connection saving (%) .................................................... 67
7.2 Posts to download when becoming online ...................................... 75
7.3 Data to download when becoming online ..................................... 76
A.1 Connection saving (%) ............................................................... 159
A.2 Connection saving (%) (cont.) .................................................... 160
# List of Figures

3.1 Number of friends for each user ................................................. 20
3.2 Photo and video posting pattern of the circles of users (log-log scale) 23
3.3 Average number of posts by circles (min, max and average per hour) 25
3.4 Percentage of online friends of users (min, max and average per minute for 15 days) ................................................................. 26
3.5 Online duration pattern of the circles of users (and its CDF distribution) 27
4.1 Sample POSN interactions ............................................................ 30
4.2 POSN initialization process ......................................................... 32
4.3 Friendship establishment process ................................................. 34
4.4 Multi-level Indexing ................................................................. 36
5.1 3G/4G download performance for different devices on different cellular providers (log-log scale) ......................................................... 40
5.2 Average throughput for different file sizes (log-log scale) .......... 43
5.3 Average time for different file sizes (log-log scale) ................. 44
5.4 Cloud and OSN comparison with different file sizes (log-log scale) . 47
6.1 POSN access tokens for Alice .................................................. 51
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2</td>
<td>The user interface for creating a post</td>
<td>57</td>
</tr>
<tr>
<td>7.1</td>
<td>Sender overhead for multimedia (log scale)</td>
<td>61</td>
</tr>
<tr>
<td>7.2</td>
<td>Friend's inter-connections for all users</td>
<td>63</td>
</tr>
<tr>
<td>7.3</td>
<td>Device-to-device exchange</td>
<td>65</td>
</tr>
<tr>
<td>7.4</td>
<td>Likelihood of finding friends online at a given time with CDF distribution</td>
<td>66</td>
</tr>
<tr>
<td>7.5</td>
<td>Help from online friends</td>
<td>68</td>
</tr>
<tr>
<td>7.6</td>
<td>Average number of posts by circles</td>
<td>70</td>
</tr>
<tr>
<td>7.7</td>
<td>Posts to download when a user becomes online (log-log scale)</td>
<td>72</td>
</tr>
<tr>
<td>7.8</td>
<td>Posts to download when a user becomes online (log-log scale) (cont.)</td>
<td>73</td>
</tr>
<tr>
<td>7.9</td>
<td>Posts to download when a user becomes online (log-log scale) (cont.)</td>
<td>74</td>
</tr>
<tr>
<td>8.1</td>
<td>Comment propagation</td>
<td>78</td>
</tr>
<tr>
<td>8.2</td>
<td>Performance comparison of embedded comments vs. comment file links</td>
<td>80</td>
</tr>
<tr>
<td>8.3</td>
<td>User managed propagation</td>
<td>81</td>
</tr>
<tr>
<td>8.4</td>
<td>Delegated comment propagation</td>
<td>83</td>
</tr>
<tr>
<td>A.1</td>
<td>Hourly number of posts by circles of User 1</td>
<td>91</td>
</tr>
<tr>
<td>A.2</td>
<td>Hourly number of posts by circles of User 2</td>
<td>92</td>
</tr>
<tr>
<td>A.3</td>
<td>Hourly number of posts by circles of User 3</td>
<td>93</td>
</tr>
<tr>
<td>A.4</td>
<td>Hourly number of posts by circles of User 4</td>
<td>94</td>
</tr>
<tr>
<td>A.5</td>
<td>Hourly number of posts by circles of User 5</td>
<td>95</td>
</tr>
<tr>
<td>A.6</td>
<td>Hourly number of posts by circles of User 6</td>
<td>96</td>
</tr>
<tr>
<td>A.7</td>
<td>Hourly number of posts by circles of User 7</td>
<td>97</td>
</tr>
<tr>
<td>A.8</td>
<td>Hourly number of posts by circles of User 8</td>
<td>98</td>
</tr>
<tr>
<td>A.9</td>
<td>Hourly number of posts by circles of User 9</td>
<td>99</td>
</tr>
<tr>
<td>A.10</td>
<td>Hourly number of posts by circles of User 10</td>
<td>100</td>
</tr>
</tbody>
</table>
A.11 Hourly number of posts by circles of User 11 ........................................ 101
A.12 Hourly number of posts by circles of User 12 ........................................ 102
A.13 Hourly number of posts by circles of User 13 ........................................ 103
A.14 Hourly number of posts by circles of User 14 ........................................ 104
A.15 Hourly number of posts by circles of User 15 ........................................ 105
A.16 Hourly number of posts by circles of User 16 ........................................ 106
A.17 Photo and video posting pattern of the circles of User 1 and 2 (log-log scale) ................................................................. 107
A.18 Photo and video posting pattern of the circles of User 3 and 4 (log-log scale) ................................................................. 108
A.19 Photo and video posting pattern of the circles of User 5 and 6 (log-log scale) ................................................................. 109
A.20 Photo and video posting pattern of the circles of User 7 and 8 (log-log scale) ................................................................. 110
A.21 Photo and video posting pattern of the circles of User 9 and 10 (log-log scale) ................................................................. 111
A.22 Photo and video posting pattern of the circles of User 11 and 12 (log-log scale) ................................................................. 112
A.23 Photo and video posting pattern of the circles of User 13 and 14 (log-log scale) ................................................................. 113
A.24 Photo and video posting pattern of the circles of User 15 and 16 (log-log scale) ................................................................. 114
A.25 Number of online friends (in %) of users (minutely) ................................. 115
A.26 Number of online friends (in %) of users (minutely) (cont.) ...................... 116
A.27 Number of online friends (in %) of users (minutely) (cont.) ...................... 117
A.28 Number of online friends (in %) of users (minutely) (cont.) . . . . . . 118
A.29 Online duration pattern of the circles of User 1 and 2 (and its CDF
distribution) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
A.30 Online duration pattern of the circles of User 3 and 4 (and its CDF
distribution) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
A.31 Online duration pattern of the circles of User 5 and 6 (and its CDF
distribution) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121
A.32 Online duration pattern of the circles of User 7 and 8 (and its CDF
distribution) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
A.33 Online duration pattern of the circles of User 9 and 10 (and its CDF
distribution) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
A.34 Online duration pattern of the circles of User 11 and 12 (and its CDF
distribution) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
A.35 Online duration pattern of the circles of User 13 and 14 (and its CDF
distribution) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125
A.36 Online duration pattern of the circles of User 15 and 16 (and its CDF
distribution) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 126
A.37 Likelihood of finding friends online at a given time with CDF distribu-
tion for User 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127
A.38 Likelihood of finding friends online at a given time with CDF distribu-
tion for User 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 128
A.39 Likelihood of finding friends online at a given time with CDF distribu-
tion for User 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 129
A.40 Likelihood of finding friends online at a given time with CDF distribu-
tion for User 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130
A.41 Likelihood of finding friends online at a given time with CDF distribution for User 5 .......................................................... 131
A.42 Likelihood of finding friends online at a given time with CDF distribution for User 6 .......................................................... 132
A.43 Likelihood of finding friends online at a given time with CDF distribution for User 7 .......................................................... 133
A.44 Likelihood of finding friends online at a given time with CDF distribution for User 8 .......................................................... 134
A.45 Likelihood of finding friends online at a given time with CDF distribution for User 9 .......................................................... 135
A.46 Likelihood of finding friends online at a given time with CDF distribution for User 10 ......................................................... 136
A.47 Likelihood of finding friends online at a given time with CDF distribution for User 11 ......................................................... 137
A.48 Likelihood of finding friends online at a given time with CDF distribution for User 12 ......................................................... 138
A.49 Likelihood of finding friends online at a given time with CDF distribution for User 13 ......................................................... 139
A.50 Likelihood of finding friends online at a given time with CDF distribution for User 14 ......................................................... 140
A.51 Likelihood of finding friends online at a given time with CDF distribution for User 15 ......................................................... 141
A.52 Likelihood of finding friends online at a given time with CDF distribution for User 16 ......................................................... 142
A.53 Connection saving when getting help from online friends for User 1 .......................................................... 143
A.54 Connection saving when getting help from online friends for User 2  
A.55 Connection saving when getting help from online friends for User 3  
A.56 Connection saving when getting help from online friends for User 4  
A.57 Connection saving when getting help from online friends for User 5  
A.58 Connection saving when getting help from online friends for User 6  
A.59 Connection saving when getting help from online friends for User 7  
A.60 Connection saving when getting help from online friends for User 8  
A.61 Connection saving when getting help from online friends for User 9  
A.62 Connection saving when getting help from online friends for User 10  
A.63 Connection saving when getting help from online friends for User 11  
A.64 Connection saving when getting help from online friends for User 12  
A.65 Connection saving when getting help from online friends for User 13  
A.66 Connection saving when getting help from online friends for User 14  
A.67 Connection saving when getting help from online friends for User 15  
A.68 Connection saving when getting help from online friends for User 16  
A.69 Posts to download when User 1 becomes online (log-log scale) ....  
A.70 Posts to download when User 2 becomes online (log-log scale) ....  
A.71 Posts to download when User 3 becomes online (log-log scale) ....  
A.72 Posts to download when User 4 becomes online (log-log scale) ....  
A.73 Posts to download when User 5 becomes online (log-log scale) ....  
A.74 Posts to download when User 6 becomes online (log-log scale) ....  
A.75 Posts to download when User 7 becomes online (log-log scale) ....  
A.76 Posts to download when User 8 becomes online (log-log scale) ....  
A.77 Posts to download when User 9 becomes online (log-log scale) ....  
A.78 Posts to download when User 10 becomes online (log-log scale) ....
Chapter 1

Introduction

Online social interactions are an integral part of our daily activity. As indicated in the Milgram’s experiment that found out six degrees of separation among any two persons, we live in a small-world [16,30]. Connecting this small-world has been an important challenge for both academic researchers and commercial entities. Connecting people in the digital world has always been a priority and was addressed in various ways starting with bulletin board systems (BBS) in the dial-up era to a myriad platforms offering Online Social Network (OSN) services today. OSNs enable frequent social interaction and expansion of knowledge or gossips. Additionally, smart phones enable continuous interaction through such digital platforms on a continuous basis.

The emergence of OSNs sparked a major reform in the information spread. From information feeds to search to social interactions, users around the world are now more deeply involved with the Internet as the user-generated content undergoes perpetual growth and expansion. A powerful aspect of OSNs is the customization of the user experience and micro-targeted information flow.

The client-server architecture of the current OSN ecosystems, however, have
raised privacy concerns. In general, users have to trust host organizations with their personal data when using OSNs. OSNs collect considerable amount of personal information about their clients and provide new services based on the collected or derived information. For instance, a provider can filter advertisements based on the user profile or the user’s circle (i.e., friends). Additionally, OSNs have predictive capabilities about the users as they continuously amass user data.

An alternative to the centralized OSN platforms is decentralized peer-to-peer architectures, which provide the user with greater control of their data. The primary challenge for a decentralized OSN architecture is to deliver the content in an efficient manner, and to have minimal overhead on the users as they maintain content for regular OSN interactions. The current decentralized OSN platforms are based on either peer-to-peer architectures [8,13,15,25,36] or hybrids [6,10,17,29,37,38,41]. In a decentralized architecture, efficient sharing and timely access of objects play a vital role. In two-way friendship OSNs, users typically access small number of objects among a vast number of others, with majority of users accessing recent objects. Additionally, updates by a user must be available to his/her friends in a timely manner regardless of whether the posting user has become offline. Peer-to-peer systems require a peer to be online to exchange data, which impacts the availability of content if the peer is offline. To overcome this issue, many decentralized OSNs utilize a distributed hash table [13,15,25,36], which could reveal interactions among users, or have the user store their data on a server [6,10,29,37,41], which is not practical for most users. Finally, even though the data is decentralized, some OSNs have centralized identity management [15]. Such central repositories enable third parties monitor friendship of the users.

In this dissertation, to address privacy concerns, we introduce a decentralized
Personal Online Social Network (POSN) ecosystem which mimics real life social interactions. In particular, we decentralize the OSN platform and give direct control of the data to the user. The distributed infrastructure removes the central authorities from the OSN allowing users to share only with their intended peers. This decentralized system ensures that interaction happens between friends and third parties cannot access user content or relationships. Note that, we assume bidirectional friendship similar to Facebook rather than one directional follower relation of Twitter and Google+.

POSN utilizes cloud storage in order to address the availability issue of peer-to-peer OSN designs. Clouds have high availability and many providers offer cloud storage for free or for a nominal fee. The introduced decentralized OSN platform utilizes mobile devices in a peer-to-peer architecture, along with the cloud storage as a highly available backup. Only encrypted data is transferred to the cloud to ensure data confidentiality. Additionally, access control is directly managed by the user via the cloud API. The proposed scheme ensures that data is accessible only to the desired friends and gives the user full control over their data. It also removes centralized identity management that exists in some of the decentralized architectures.

Cloud storage is integral to efficient operation of POSN. Cloud, however, is utilized only for storage of encrypted content so that cloud provider has no understanding of the content. We assume there is no execution in the utilized cloud. Additionally, cloud complements the phone-to-phone interaction between users to efficiently distribute user content. While peer-to-peer systems require a peer to be online for data exchange, we rely on cloud to greatly enhance data availability. Finally, free cloud storage services allow POSN to function through user clients such as smart phone or tablet apps with no infrastructure of its own. A major challenge, we focus in this
project is to balance between technical efficiency of a solution versus social restrictions that exist in our daily life.

Main contributions of this dissertation are:

(i) Analysis of activity in a user’s circle to obtain a better understanding of the interactions in OSNs and data dissemination through the user’s circle (Chapter 3),

(ii) A new privacy preserving decentralized personal OSN architecture, where users interact through mobile devices and are in charge of their data by providing access control tokens for encrypted content stored in cloud (Chapter 4),

(iii) Measurement of the performance of popular social networks (i.e., Facebook and Google+) and cloud providers (i.e., Dropbox, GoogleDrive, OneDrive, Mediafire, and CopyCloud) to compare their communication efficiency in delivering various size content to the users (Chapter 5),

(iv) A fine grained access control mechanism that allows secure data dissemination in a decentralized environment (Chapter 6),

(v) Algorithms to provide efficient data dissemination through the user’s circle (Chapter 7), and

(vi) A discussion on how to exchange small objects like comments, that builds the interactions among friends (Chapter 8).
Chapter 2

Decentralized Online Social Networks

In this chapter, we present an overview of social networks followed by online social networks that have decentralized architectures.

2.1 Overview on Social Networks

The Internet has spawned different types of information sharing systems, including the Web. Recently, online social networks have gained significant popularity and are among the most popular sites. For instance, Facebook with 1.2 billion users [1], Google+ with more than 540 million users [2], Linkedin with more than 259 million users [2], and Twitter with more than 230 million users [5], are the most popular sites built on social networks.

A social network consists of a user profile for each participant, his or her social links, and a variety of additional services. Social network sites are web-based services
that allow individuals to create a public profile, to create a list of users with whom to share connections, and view and cross the connections within the system.

In basic structure of an online social network, a user profile contains relations and resources. Relations include friends, family, etc. Resources include private information, private messages, wall status, pictures, videos, etc. To participate in an online social network, users register with a site, possibly under a pseudonym. Some sites allow browsing of public data without explicit sign-up. Users may share information about themselves (such as birthday, place of residence, likes and interests), which is added to the user’s profile.

The social network is composed of user accounts and links between users. Some sites allow users to link to any other user, without consent from the link target. Other sites (such as Facebook and LinkedIn) require consent from both the initiator and the target before a link is created connecting the users. The users connected by a link can be real-world acquaintances, business contacts, can share an interest or have any other reason. User links in social networks can serve the purpose of both hyperlinks and bookmarks in the Web. A user’s links, along with her profile, are visible to those who visit the user’s account. Thus, users are able to explore the social network by following user-to-user links, browsing the profile information and any contributed content of the visited user. However, social networks often have some access control constraints as who can access wall posts, pictures, videos and etc. For example, LinkedIn only allows a user to browse other user accounts within her neighborhood (e.g., a user can only view other users that are within two hops in the social network).

Most sites enable users to create and join special interest groups. Users can post messages to groups and upload shared content to the group. Certain groups are
moderated; admission to such a group and postings to a group are controlled by a user designated as the group’s moderator. Other groups are unrestricted, allowing any member to join and post messages or content.

The most popular online social network Facebook has 1.2 billion users where on average 728 million people become online at a given day. Most of the users use a mobile device, i.e., 874 million people, to access the Facebook platform. On average, a user spends 20 min per day where they post 422 million updates, 200 million photos, and 734 thousand comments. Moreover, on average, a user has 130 friends (while the number is 214 in the United States) [26] who are divided into 12 groups [40].

2.2 Decentralized Online Social Networks

2.2.1 Cachet

Cachet [36] is a decentralized architecture for social networks that primarily deals with privacy issues of user content. It combines decentralization, attribute-based encryption, and the use of caches to provide high availability, low latency, and flexible policies for protecting data. Cachet uses a distributed pool of nodes to store user data and ensure availability. Storage nodes in Cachet are untrusted; hence, cryptographic techniques such as attribute-based encryption to protect the confidentiality of data is leveraged.

Central to Cachet is a hybrid structured-unstructured overlay in which a conventional distributed hash table is augmented with social links between users. This helps efficient dissemination and retrieval of data.

In addition to distributed hash table, a gossip-based social caching algorithm is added to the system. New updates are immediately propagated to online social
contacts. When an offline user comes back online, a presence protocol is used to locate online contacts and query them directly for updates. The DHT is then used to retrieve updates that may not be cached, ensuring high availability of data.

Although Cachet stores messages at untrusted nodes with sufficient encryption and these nodes do not have access to message, they can detect the presence of information. Also, this network could be exposed to man-in-the-middle attacks if the entire contents of the message are replaced. A better mechanism will be to ensure information not stored at untrusted nodes in the first place. Since there is an implementation of a caching layer, Cachet will do better in leveraging it. We feel a better mechanism would have been to ensure sufficient replication at trustworthy nodes only and to ensure these nodes do not have access to information unless authorized.

2.2.2 Persona

Persona [10] is another decentralized social network that offers flexible and fine-grained access control for user data by combining attribute-based encryption with traditional public-key cryptography. Users are identified by public keys they exchange out of band while creating OSN links. Data confidentiality and privacy is assured through encryption. Users have to trust an extension to interact with Persona and can create multiple identities. It is not built upon a DHT; and hence, users and applications need a storage service hosted on a dedicated storage server or a user’s own storage server. The storage service authenticates write operations through the requester’s public key and therefore can learn the user’s social contacts.
2.2.3 Safebook

Safebook is one of the systems that adopts a decentralized architecture relying on the cooperation among the users of online social network application [14, 15]. The nodes in Safebook form two types of overlays. One is a P2P substrate, which provides lookup service. The other is a set of structures called matryoshkas, which in the social network layer provides data storage and communication privacy created around each other. The purpose of this structure is to ensure that the data and identity of a node is closely guarded secret within the matryoshka. The matryoshkas, similar to Tor relays, provide privacy and ensure communication is anonymous.

Although crucial issues related to privacy are addressed, the design of the network is not responsive enough. We feel that the introduction of multiple layers of nodes surrounding a node is not required and it becomes unwieldy to manage user data. To have a responsive system, at least three to four shells around the matryoshka are required. However, using multiple shells of nodes increases the attack surface and can make it easy to compromise information through any of the nodes. If a circle of trusted nodes is built with correlations of online/offline patterns, we can reduce the number of shells required and reduce the attack surface.

2.2.4 Peerson

Peerson [13] is project that aims to build a privacy preserving P2P OSN. It aims at keeping the features of OSNs while overcoming two limitations: privacy issues and the requirement of Internet connectivity for all transactions. Peerson has a two-tiered architecture; where lookup of users using a DHT is taken care in one tier and storage of information is done in the second tier. The main properties
of Peerson are encryption, decentralization, and direct data exchange. Encryption provides privacy to the users, and decentralization based on a P2P infrastructure provides independence from third party OSN providers. Decentralization makes it easier to integrate direct data exchange between user devices into the system. Direct exchange also allows users to use the system without constant Internet connectivity, leveraging real-life social networking and locality.

It is mentioned that without internet connection the messages can be transmitted via this design. This feature is primarily useful in a home scenario between users on a LAN. However, users on such a network can already share data via existing file sharing or messaging services which does not require using an OSN network. Introducing a new layer between users to transmit data will introduce unnecessary delays. Hence, this feature may not appeal to a broad user base.

2.2.5 Diaspora

Diaspora is a distributed social network that is currently used by around quarter million people operating from 141 different domains [6]. The main premise of Diaspora is to create a social network where data is controlled by the people without a central entity that can aggregate information shared by everyone on the network. To make this a reality, Diaspora uses a pod based design. At any given time, multiple pods can exist on the network. Other users can create accounts on a particular pod they trust and connect with others who can be anywhere on the network. A user also has the flexibility to migrate his/her data between pods which ensures that there can be no lock-in with a single provider.

Diaspora allows a user to share information with only a subset of people in his/her network or the entire public.
There is however a few shortcomings of the Diaspora network such as lack of tools to migrate data between pods. From privacy aspect, all the data that is being transferred is encrypted using SSL that prevents outside eavesdroppers from accessing data. However, when a user shares a post with a user on another pod, his/her post and personal information is replicated at the other pod. This makes the user lose control over his/her data which may not be deleted since the pod is not under his/her control. Most important issue with Diaspora system is that the pod operator has control over all information, public and private, that users store in a pod. This also can result in a security threat for a user registered in a pod if the operator has malicious intents. Hence, rather than a single entity, Diaspora puts user data on multiple entities.

2.2.6 My3

The main motivation for the design of My3 is to build a decentralized OSN by leveraging trust of nodes to store information [33]. The main advantage of the My3 network is to lower replication factors for real world churn rates. Its main drawback, however, is that the users need to define their own trusted nodes, which store unrestricted user information.

The major issue with My3 is requirement of users finding trusted nodes. The user has the responsibility to select trustworthy nodes, which might end with a wrong decision. Another weakness of this design is the assumption that a node is trusted forever. The control over a data can be lost if there is any change in the trustworthiness of a node.

In the age of Moore’s law where the price of storage falls over the time, considering such a huge trade off for lesser replication factors is not ideal.
2.2.7 Tent

Tent [7,24] creates a distributed social network by building a protocol that all the app developers and service providers adhere to and thereby giving back the user power to choose service providers and applications that s/he can trust and migrate data easily without much overhead. Users keep data in a place they control, select apps based on the functionality that they need, and use the protocol for interactions with other users on the network. Tent is a protocol that servers use to interact among themselves. It requires users to have personal servers to host their data. After creating an account at a Tent server, users utilize Tent applications to interact with others. They create posts using applications and the applications in turn transfer the posts to the user’s server. The server then saves a copy for archival purposes and transfers a copy to servers of other users who have subscribed to the user’s posts. These tent servers then forward the posts to application that these users have designated to handle the content of the posts. While, Tent provides a protocol for decentralized interaction, it does not address storage needs.

2.2.8 LifeSocial

LifeSocial [21] creates a distributed network built only with peers that contribute resources to the network without using servers as intermediaries. Access rules are implemented to ensure the user’s stored data on the network can only be accessed by the intended peers. LifeSocial is designed with the main premise to leverage existing and proven components, and to create a modular plugin-architecture to assure extensibility. It is assembled using FreePastry [4], a structured peer-to-peer overlay for data storage, and PAST [3] to achieve reliable replication of the data. LifeSocial
implements its own access control scheme on top of these components. The system relies on plugins for general OSN interactions such as profile management, friend management, group management and, photo albums.

LifeSocial is still under closed beta testing and not open to public.

### 2.2.9 Gossple

The concept of an anonymous social network that helps users in building new relationships with other users in the network based on mutual interests that they share between them is discussed in this study [11]. Discovery of new links is accomplished with all the users gossiping between them about their mutual interests. Each user shares gossip digests of their interests and also computes the distances with others in the network to automatically infer personalized connections in the entire network. All this is accomplished in a low-bandwidth environment which does not require too many resources.

One of the challenges of this system is to measure the similarity of nodes and find the closest node that is similar to any node sharing same interests. Another challenge is the shared data of a user should not be stored in a centralized server; instead a decentralized approach should be used.

### 2.2.10 PAC’nPost

PAC’nPost [9] is a framework to implement a micro-blogging social network with an unstructured P2P overlay. The basic goal is to provide the ability for users to publish content, follow and search for other users.

The retrieval mechanism is based on a probably approximately correct (PAC)
search architecture in which query is sent to a fixed number of nodes in the network. It is a way to predict the probability of a successful search based on the distribution of documents in the network. An overall retrieval accuracy alpha is calculated by using the number of documents retrieved with PAC search as compared to an exhaustive search performed on the whole network. Retrieval accuracy is a useful metric to determine how PAC performs with regards to an exhaustive search.

For replication of data to the required number of nodes, gossip protocols are suggested. Gossip algorithms normally use to find efficient ways to disseminate a piece of information across the whole network. This is useful in cases where a particular bit of information needs to reach a wide audience quickly. Document spreading algorithm described in this study is designed such that a document has a high probability of being copied to only required number of nodes.

2.2.11 Vegas

Vegas [17] is a secure and privacy preserving p2p online social network. The scope of Vegas network is to maximize security of data and privacy for users at the cost of functionality. In Vegas, each user holds a separate public/private key pair for each of his friends in the network. This ensures that all the communications between any two links in the network stay private only between communicating parties. To ensure availability of data at all times even when the users are offline, data stores which are user-writable and world-readable space are used. A user can utilize multiple data stores to increase the degree of decentralization.

While data is encrypted, user interactions are visible to everyone. Hence, Vegas does not protect privacy of an important component of OSNs, namely interactions between users. In this sense, it is weaker than even centralized systems where a
limited number of entities can observe user interactions.

2.2.12 Pythia

Pythia [35] is an attempt to present a privacy aware P2P network for social search. Social search is a technique used to identify other users on the network who are knowledgeable in the areas they have questions in. Pythia uses "controlled flooding" of queries to enhance privacy in social search. Every user in the network maintains connections with other friends and a list of declared expertise areas. These may be private and are not revealed to other users in the network. Nodes in the network can be online/offline/idle and can answer or route questions depending on their current status. A user creates a query with a set of tags of expertise areas required to answer the question. The query is then routed to an available expert and the answer is returned back. To ensure anonymity of the user who asks and answers the question, Pythia splits the entire network into smaller communities called "flood zones" and propagates the queries locally by sending it to experts only within the community. There is also always constant chatter of dummy questions and answer in the background to ensure an attacker cannot determine the user answering the actual question similar to anonymizers.

2.2.13 Lockr

Lockr [39] improves privacy for centralized and decentralized online content sharing services by reducing the chances of mismanagement or accidental disclosure of social networking information. Although Lockr separates the management of social relationships and shared content, it is still possible to map content that is shared among
ordinary OSNs to anonymous OSN identities. The content itself is also stored in centralized OSNs, i.e., informational self-determination cannot be achieved.

2.3 POSN compared to Decentralized OSNs

As can explained in the previous section, there have been many studies that explore decentralized OSNs using different storage and encryption schemes in order to improve user privacy. The primary concern for decentralized OSNs is how the data is stored and how access control is achieved. Table 2.1 present a summary of privacy aware OSNs in terms of services provided, system architecture and security. In particular, it indicates which OSN services they support (i.e., micropublishing, commenting, multimedia sharing, news feed, and instant messaging), their system architecture (i.e., whether they are scalable, how the data is stored and availability mechanisms) and how they address security services (i.e., authentication, integrity, confidentiality, and anonymity).

PeerSon [13] utilizes a Distributed Hash Table (DHT) to look up where the data is stored and uses encryption for fine-grain access control. Safebook [15] also uses a DHT and encryption, as well as trusted parties that serve as mirrors to help distribute the data. Safebook also ensures anonymity from observers outside of the friend network by using a multi-hop system, but the users within a specific group do not have anonymity from each other.

Several studies incorporated Attribute Based Encryption (ABE) into decentralized OSN platforms in order to ensure data confidentiality and fine-grained access control. Cachet [36] improves the architecture of Decent [25] as the users store data container objects in a DHT for determining access control. One issue is that the
access policy is defined openly in the container, and this allows for all users to observe it. In addition, it uses caching to improve the availability of data. Lotusnet [8] utilizes DHT for communication and identity management.

PrPI [41] uses the decentralized OpenID management system with fine grained privacy controls, so that users can use their established personas in accessing the data. Diaspora [6] is based on a client-server model where every user has his own server, pod, which is used for storage, communication and access control. SuperNova [38] relies on Super peers that have more bandwidth, more storage space and higher availability. Super peers actively participate in the control infrastructure of the system, providing lookup service, storage service, and tracking users’ replicas.

Finally, there has been several decentralized OSN designs that utilize a hybrid of client-server and peer-to-peer architectures. Vis-a-Vis [37] assigns each user their own personal virtual server (i.e., VIS) to store data. Each VIS is treated as a member in an overlay network that represents different social groups. Confidant [29] stores user data without encryption on secure devices. It also relies on trust relationships with friends so they can replicate the data to have better availability. Vegas [17] utilizes trusted servers to increase data availability for the users, and the data is encrypted using symmetric keys.

POSN, incorporates popular OSN functionalities from commenting to instant messaging in a decentralized architecture among mobile devices. The system uses cloud storage to store and disseminate data, which enables the friends access data even if the data owner is offline.
Table 2.1: Comparison of decentralized OSNs

<table>
<thead>
<tr>
<th>OSNs</th>
<th>Micropublishing</th>
<th>Commenting</th>
<th>Multimedia Sharing</th>
<th>Newsfeed</th>
<th>Instant Messaging</th>
<th>Stability</th>
<th>Storage</th>
<th>Availability</th>
<th>System Architecture</th>
<th>Security Services</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OSN Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Authentication</td>
<td>Integrity</td>
<td>Confidentiality</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Anonymity</td>
</tr>
<tr>
<td>Salebook [15]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✗</td>
<td>DHT</td>
<td>Replication</td>
<td>TIS</td>
<td>Private key</td>
<td>Encrypted message</td>
<td>Group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decent [25]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✗</td>
<td>DHT</td>
<td>Replication</td>
<td>Not addressed</td>
<td>ABE &amp; Signature</td>
<td>Encrypted message &amp; storage</td>
<td>All</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cachet [36]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✓</td>
<td>DHT</td>
<td>Replication &amp; Caching</td>
<td>Not addressed</td>
<td>ABE &amp; Signature</td>
<td>Encrypted message &amp; storage</td>
<td>All</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lotusnet [8]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✔</td>
<td>DHT</td>
<td>Stable Nodes</td>
<td>OpenID</td>
<td>Keys</td>
<td>Encrypted message</td>
<td>Individual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrPT [41]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✓</td>
<td>Trusted servers</td>
<td>Stable nodes</td>
<td>OpenID</td>
<td>Certificate</td>
<td>Encrypted storage</td>
<td>Individual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diaspora [6]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✓</td>
<td>User pods</td>
<td>Stable nodes</td>
<td>Not addressed</td>
<td>PGP / Keys</td>
<td>Encrypted message</td>
<td>Group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SuperNova [38]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✗</td>
<td>Super peer</td>
<td>Replication</td>
<td>Not addressed</td>
<td>Not addressed</td>
<td>Encrypted storage</td>
<td>All</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vis-a-Vis [37]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✓</td>
<td>Virtual ind. servers</td>
<td>Stable nodes</td>
<td>Not addressed</td>
<td>Not addressed</td>
<td>Encrypted message</td>
<td>Group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confidant [29]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✓</td>
<td>Trusted friends</td>
<td>Replication</td>
<td>Not addressed</td>
<td>Certificate</td>
<td>Encrypted message</td>
<td>Group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vegas [17]</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✗</td>
<td>Data stores</td>
<td>Stable nodes</td>
<td>Not addressed</td>
<td>Public Key</td>
<td>Encrypted message</td>
<td>Not addressed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POSN</td>
<td>✓ ✓ ✓ ✓ ✗</td>
<td>✓</td>
<td>Cloud</td>
<td>Cloud</td>
<td>Email, SMS</td>
<td>Keys</td>
<td>Encrypted message &amp; storage</td>
<td>Individual &amp; group</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: ✓: Supported, ✗: Not supported, ✔: Optional feature, ✗: Not applicable
Chapter 3

Measurement of User Interactions

When designing a new OSN platform, it is critical to understand how users interact over the OSN platform in order to provide crucial communication mechanisms and distribute content efficiently. In this section, we analyze online activities that happen around a user, i.e., circle. We developed a Facebook application to measure the activities of users and their circles (i.e., posts shared by their friends) [19].

We collected the information from volunteers through the Facebook API. Each volunteer explicitly allowed the data collection app in their Facebook account. We obtained statistics of every post shared in the user’s circle, for volunteers that provided continuous measurements. We set the application to collect the volunteer’s wall every hour and collect friends’ online statuses every minute. We recorded the number of friends, the number of friendlists (and number of friends in each group), the number of posts made within an hour and their size, the number of likes and comments for each post, the number of messages sent thorough chat, and the online duration of the friends of the users. We did not store any identifying information or content during the measurement period or even record who volunteered for the measurements, as we
used random identities for users and their friends.

### 3.1 Facebook Measurements

We asked a number of Facebook users to voluntarily participate in our study through email and Facebook posts and 18 users accepted to participate in the study. Using the app, we obtained measurements of 18 Facebook users and their circles (i.e., posts shared by their friends) for 15 days in June 2014. Two of the users had almost no activities in their friend circle during measurement period, so we dropped these two users and did not include their measurement in the results.

We also observed whether the friends of the monitored users are connecting to the Facebook. Through instant messaging service, we detected whether a friend is online or offline and whether they connected through a mobile device. A user also has the option to appear offline, even if he/she is logged in to her/his Facebook account. Figure 3.1 presents the average number of friends using a mobile device and a PC. We marked a user as using mobile device if at any time s/he appeared to connect via a mobile device. Note that, some of the friends were never online or appeared to be
offline during our measurements.

The number of the users monitored in this study might seem to be small at first glance. Since our study is volunteer based, we only recorded activities of volunteers who installed the measurement app in Facebook. Especially in social studies where it is important to have the willingness of study participants, the number of the volunteers are typically few.

Among analyzed users, on average, the number of friends is 220 where 146 of whom used a smart device. The most number of friends for volunteers is 635, and the least number of friends is 25. The average number of groups a user creates for her/his friends is 9. Considering all friends of 16 users, there were a total of 3,523 friends and 27.3% never seem to be online during our measurement period. Also, 76.3% of online friends used a mobile device.

Table 3.1 shows total number of posts shared by users and their circles over 15 days. On average; the number of friends is 220 (146 of whom used a smart device) and the number of groups is 9. Similarly, per day, average number of text posts in the users’ circle is 53, links is 37, pictures is 51, videos is 7, and chat messages is 85. The monitored users themselves uploaded just a total of 25 pictures and 2 videos over the same period.

### 3.2 Posting Patterns

In this section, we present the statistics of the multimedia and non-multimedia posts generated by users’ circles. Figure A.24 shows how multimedia (i.e., photo and video, respectively) posts are generated by the circles of each user. Each point in the figures indicate one post and is ranked by file size. Multimedia posts generated for
Table 3.1: Posts by the user and user's circles

<table>
<thead>
<tr>
<th>User</th>
<th>Friends</th>
<th>Videos</th>
<th>Photos</th>
<th>Links</th>
<th>Status</th>
<th>Comments</th>
<th>Videos</th>
<th>Photos</th>
<th>Links</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>635</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>15</td>
<td>21</td>
<td>425</td>
<td>3,145</td>
<td>3,023</td>
<td>2,349</td>
<td>6,219</td>
</tr>
<tr>
<td>User 2</td>
<td>495</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>15</td>
<td>17</td>
<td>97</td>
<td>1,247</td>
<td>1,041</td>
<td>2,426</td>
<td>4,513</td>
</tr>
<tr>
<td>User 3</td>
<td>449</td>
<td>0</td>
<td>3</td>
<td>10</td>
<td>13</td>
<td>49</td>
<td>173</td>
<td>814</td>
<td>670</td>
<td>1,020</td>
<td>3,896</td>
</tr>
<tr>
<td>User 4</td>
<td>435</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>16</td>
<td>9</td>
<td>190</td>
<td>756</td>
<td>477</td>
<td>1,020</td>
<td>3,334</td>
</tr>
<tr>
<td>User 5</td>
<td>290</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>162</td>
<td>841</td>
<td>411</td>
<td>787</td>
<td>2,910</td>
</tr>
<tr>
<td>User 6</td>
<td>205</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>9</td>
<td>25</td>
<td>113</td>
<td>720</td>
<td>446</td>
<td>686</td>
<td>2,527</td>
</tr>
<tr>
<td>User 7</td>
<td>173</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>7</td>
<td>31</td>
<td>116</td>
<td>510</td>
<td>428</td>
<td>651</td>
<td>1,494</td>
</tr>
<tr>
<td>User 8</td>
<td>141</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>30</td>
<td>405</td>
<td>233</td>
<td>586</td>
<td>1,341</td>
</tr>
<tr>
<td>User 9</td>
<td>135</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>11</td>
<td>62</td>
<td>443</td>
<td>286</td>
<td>483</td>
<td>1,679</td>
</tr>
<tr>
<td>User 10</td>
<td>129</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>69</td>
<td>999</td>
<td>530</td>
<td>509</td>
<td>2,986</td>
</tr>
<tr>
<td>User 11</td>
<td>116</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>17</td>
<td>19</td>
<td>82</td>
<td>549</td>
<td>477</td>
<td>559</td>
<td>1,718</td>
</tr>
<tr>
<td>User 12</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>55</td>
<td>483</td>
<td>123</td>
<td>169</td>
<td>1,574</td>
</tr>
<tr>
<td>User 13</td>
<td>78</td>
<td>2</td>
<td>15</td>
<td>6</td>
<td>39</td>
<td>97</td>
<td>52</td>
<td>329</td>
<td>225</td>
<td>478</td>
<td>1,173</td>
</tr>
<tr>
<td>User 14</td>
<td>69</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>18</td>
<td>43</td>
<td>84</td>
<td>188</td>
<td>879</td>
</tr>
<tr>
<td>User 15</td>
<td>48</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>16</td>
<td>28</td>
<td>232</td>
<td>128</td>
<td>223</td>
<td>751</td>
</tr>
<tr>
<td>User 16</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>9</td>
<td>44</td>
<td>747</td>
<td>286</td>
<td>530</td>
<td>1,933</td>
</tr>
<tr>
<td>Average</td>
<td>220</td>
<td>0.1</td>
<td>1.4</td>
<td>2.8</td>
<td>9.3</td>
<td>19.9</td>
<td>107.3</td>
<td>766.4</td>
<td>554.3</td>
<td>791.5</td>
<td>2,433</td>
</tr>
<tr>
<td>Daily A</td>
<td>-</td>
<td>0.006</td>
<td>0.093</td>
<td>0.19</td>
<td>0.6</td>
<td>1.3</td>
<td>7.2</td>
<td>51.1</td>
<td>47</td>
<td>52.8</td>
<td>162.2</td>
</tr>
</tbody>
</table>

Each user’s circles are presented individually in Appendix A. Considering photo file sizes, five-number summary statistics is; min = 1.28 KB, first quartile = 32.85 KB, median = 49.04 KB, third quartile = 67.73 KB, and maximum = 896.06 KB. Similarly, for the videos; min = 68.47 KB, first quartile = 3.36 MB, median = 8.21 MB, third quartile = 20.15 MB, and maximum = 185.64 MB. We observe that few photos are larger than 100 KB. Also, none of the posted photo is greater than 900 KB as Facebook compresses the uploaded photos. Similarly, most of the videos are smaller than 10 MB, but exceptionally there are videos larger than 100 MB. Considering all multimedia content the median file size is 51.5 KB while average file size is 1.38 MB indicating large files are outliers.
We also analyzed how active the friend circles were when posting to Facebook, by using the activities of users and their circles. *Per day*, the average number of comments in the users’ circle is 162, status updates is 791, links is 47, pictures is 51, videos is 7, and chat messages is 85. The monitored users themselves uploaded just a total of 25 pictures and 2 videos over the 15 days.

Figure 3.3 presents the average number of posts (i.e., videos, photos, links, status updates, and comments) shared by the circles of all users *per hour*. The blue
points show the average number of posts per hour, the error bars show the min and max values for the hour, and the red line shows the average of averages. In total, friends of the 16 monitored users posted 38,927 comments, 12,664 status updates, 8,868 links, 12,263 pictures, and 1,716 videos over the 15 day period. As expected, non-multimedia posts are shared the most compared to multimedia content. Also, the number of posted pictures is much larger than the number of videos. The figures for each user separately can be found in Appendix A.

3.3 Online Pattern

Through instant messaging chat service of Facebook, we can detect whether a user or her/his friends are online or offline at a given time. We monitored the online pattern of the users’ circle through this feature of Facebook. We monitored and logged the friends’ online pattern every minute. Note that, the users that mark themselves to be offline cannot be detected, and they would be perceived as offline in our measurements. Hence, the online measures are an underestimation of the online presence.

Figure 3.4 shows the average online ratio of the 16 users’ friends every minute. Individual figures for each user, that shows the percentage of online friends with minute intervals can be found in Appendix A. We only present the measurements when all 16 users had allowed the measurement app at the same time and the prior and later data from a subset of users was ignored. The graph starts from the time when all the user data was available and ends at the end of the 15th day. It shows at each minute interval what percentage of a user’s friends were online. The blue points show the average ratio of online friends, the error bars show the minimum and maximum percentage of online friends, and the red line is the average of averages.
Figure 3.3: Average number of posts by circles (min, max and average per hour)
On average, 16.3% of the friends are online, and the user may directly interact with these friends.

Figure 3.4: Percentage of online friends of users (min, max and average per minute for 15 days)

We also analyzed the probability of a friend being online for 16 users. Figure A.52 presents the probability distribution function for the online timing of user’s circles (along with CDF of the distribution). This figure ranks friends based on their online duration, and a higher percentage indicates a higher likelihood of seeing that friend online. Each line presents the percentage of online duration of a friend of the 16 users. None of the users’ friends was always online during the 15 days of observation. Note that the figure ignores the friends that never appeared to be online, which means they are either passive users of Facebook or they hide their presence. Hence, these results are an underestimate of user’s circles as some users might have chosen to appear offline for chat messaging in Facebook.
3.4 Facebook Usage Analysis

We observe that measured Facebook users had a small friend circle (i.e., 220 friends on average) while others have claimed a larger average (such as 352 in [22]). As proposed OSN focuses on personal interactions, the number of friends would be lower for POSN. Anthropologists indicate a person has a limited number of relations that s/he can maintain. For instance, [23] claims humans have meaningful regular interaction with approximately up to 150 individuals. Hence, we can build a decentralized platform around a user where s/he interacts with friends through her/his device(s) without a central server that aggregates the data around her/his friend circle in a decentralized manner.

We also observe that majority of posts in a user’s circle are non-multimedia posts that can be efficiently disseminated among mobile devices. As only 9% to
16% of a users friends are online at a given time, on average the user can directly
send the non-multimedia post to online friends. Similarly, a user can send photos to
her/his online friends without much bandwidth consumption but it should rely on
cloud to send multiple copies of videos, even though they are rarely posted. Recall
that majority of photo posts are less than 100K (see Figure A.24) and sending to
online friends would consume about 2.7 MB (considering 16.3% of 220 friends are
online on average). Alternatively, the user can rely on cloud to disseminate photos,
as well, to further conserve data charges and power or utilize a structured multicast
to the online users interested in the photo.

Additionally, as about a third of a users’ friends are online more than half of
the time (see Figure A.52), those friends can help disseminate dynamic content (such
as comments in Chapter 8) to reduce look up overhead of friends when the user is
offline.
Chapter 4

Personal Online Social Network (POSN)

Interactions between the user and their friends and groups is a vital part of the social network experience, and must be incorporated into a decentralized OSN platform. These interactions also need to be carried out as efficiently as possible to reduce the overhead for users. In this chapter, we present a privacy-preserving decentralized OSN platform that utilizes cloud storage and mobile devices. The Personal Online Social Network (POSN) gives the user full control over their data, as content is only shared with the intended peers directly by the user [18].

4.1 System Overview

One of the main challenges for decentralized peer-to-peer social networks is the availability of the content. To address the data availability issue, we utilize storage clouds (such as Dropbox, GoogleDrive, and OneDrive) and/or the user’s personal computer
(PC) to distribute user content. The cloud is only used to store encrypted data so that cloud provider has no understanding of the user content.

Mobile devices are also utilized to provide the computation power to manage the data and access tokens. Many cloud providers provide APIs to seamlessly connect mobile applications directly to their cloud services. Pairing the mobile device with the cloud allows for efficient distribution of user content, and provides the OSN functionality without any additional infrastructure for POSN. This also provides anonymity against an oppressive regime that monitors or blocks certain apps/servers as storage clouds are popular and POSN can utilize any cloud storage provider. The user grants access to her/his friends, giving the user full control over what content is shared and who can see the content.

Figure 4.1 shows sample interactions in the POSN system. In this example, Alice is connected to her friends: Bob, Chris, David, and Eric. When Alice wants to post a new content, she will encrypt the data and upload it to her cloud. Alice’s
online friends will receive a notification with a key for multimedia content so that they can directly fetch the data from Alice’s cloud or directly receive non-multimedia posts while a copy is uploaded to the cloud for offline friends. Furthermore, a user can have a PC with or without cloud backup. If Eric posts a new content, he can upload data to his PC, which in turn uploads it to the cloud. When Alice receives a notification from Eric, she can fetch the data from Eric’s cloud or PC. A user can also choose to use a PC without any cloud backup, for example David, who is offline. The PC benefits the POSN platform by processing the data even if the user is offline in addition to storage. One drawback is that the PC without a cloud backup would affect data availability if it goes offline as typical user gets a dynamic IP from their ISP. If such PC users obtain a static IP, they do not need to have a cloud account for high availability.

### 4.2 Generating Unique IDs

To allow for searching and updating content efficiently in POSN, globally unique IDs are needed for users, groups, posts, and comments without a central repository or global coordination. The POSN platform currently relies on SHA hashing algorithm along with different input strings to generate the unique IDs. The input strings for the different IDs are as follows:

- **Friend ID** - email address plus a random salt
- **Group ID** - ID of the user, name of the group, and the time created
- **Post ID** - ID of the user, and time created
- **Comment ID** - user ID, post ID, and the time created
These pseudo IDs could be used to identify content in friend circle while interaction between individual friends could be obfuscated from other friends.

4.3 System Initialization

Figure 4.2 presents sample snapshots of the POSN app in development (available at https://github.com/posn/POSN-app [28]). When the application is launched for the first time, the user provides their email address and a password. The password is used to authenticate the user every time the app is opened, and is used to create a symmetric key to encrypt/decrypt files that are stored on the device. It is important to note that the password itself is not stored on the device or in the cloud. The user also creates a public/private key pair that is randomly generated from a seed that is obtained from a box that the user draws in as shown in the figure.
The user is prompted to initialize a cloud account, and is required to either create a new account with a cloud provider or sign in with an existing account. The final initialization step is to create different friend groups and send out initial friend requests. The user can organize their friend lists into multiple groups, where each friend can be in one or multiple. Each friend group is given a unique symmetric key, and this key is used to encrypt the group’s wall post file.

4.4 Friendship Establishment

Establishing friendship in a decentralized system requires methods to find friends and ensure their identities. As there is no central database, we cannot utilize friend recommendation systems such as [34]. Since the user is using a mobile device, the phone’s contacts can be used to add new friends within POSN. An email message or SMS is sent to the desired user, along with a Uniform Resource Identifier (URI), that contains relevant information to initiate the request. As POSN is intended for close interactions, this approach should be sufficient to add friends.

The friendship establishment process has three phases as shown in Figure 4.3. The requesting user (e.g., Alice) first sends an email to the new friend (e.g., Bob) that contains a URI with the following: user ID, public key, temporal cloud URL, nonce. The temporal cloud URL is used to ensure that a man-in-the-middle attack is avoided, where an adversary, including a malicious email provider, could intercept the initial email and respond back to Alice with their information instead of Bob’s.

When the friend opens the email and the URI is fetched, it will open the POSN application. In the application, Bob can either accept or decline the friend request. If the friend request is accepted, an email is sent back to Alice with the
It is important to note that the information sent in the Bob’s URI is encrypted with Alice’s public key to ensure that only Alice can read the information. Bob will select which group Alice will be a part of and create a new friend entry for Alice.

Once Alice receives Bob’s email accepting her request, Alice creates a new friend entry for Bob and adds the appropriate information to the file. The direct link to the friend file is encrypted with Bob’s public key and sent as a direct message to Bob’s phone. The new encrypted link to the friend file is also uploaded to a temporal file in the cloud in case Bob is offline. Even if an adversary knew about the temporal file, the encryption on the data prevents the adversary from viewing the data inside the temporal file without knowing Bob’s private key. Once Bob becomes online, he will notify Alice that the temporal file has been fetched and Alice can delete the file.
4.5 Search Optimization

One of the features of the OSNs is the efficient search for objects. Typically people are restricted with their and friends’ multimedia content. Multimedia content is also searched through the tag information that the owner placed at the time of uploading the content or added later by friends.

In order to support content search, multimedia content in POSN is tagged by the owner at the time of creation. The tag information may include the following: (i) time, (ii) location, (iii) person, and (iv) general tags.

Searching is not a straightforward process in decentralized systems since there is no central database in the system. In POSN, the cloud is used only for storage purposes and friends content is scattered across several locations. In order to search for a content among friends, the wall files from all friends should be downloaded to a client to be searched through. Such a scheme is very inefficient since the number of multimedia content belonging to a user can be very high. Furthermore, this search is not limited only with the owner’s content, it should include the content of each friend. It is prohibitive to download all these content, decrypt them, and search for desired content.

In order to overcome this problem in POSN, an index structure is implemented as described in Figure 4.4. Whenever a new multimedia content is created, its tags are inserted into the index file by the content creator and uploaded to the cloud along with the post.

Considering the POSN structure of framework which supports grouping, one index file will not be enough to handle different groups. Because an owner might post a multimedia content to a specific group, inserting it’s tag information into a common
index will hint other users of its existence. Hence, POSN keeps a different index file for each group encrypted with the symmetric key of the group that it belongs to.

Finally, a user would need to download index files from each of her/his friends. To improve search efficiency, POSN preemptively processes the index files of all friends and create a new index structure on the client whenever the user is online. These aggregate indexes can be shared with community members, i.e., group of friends that are tightly connected, to save computing.
4.6 Anonymity

A privacy concern POSN is susceptible to is listening attacks, where an adversary, including the cloud provider, can monitor the traffic going to and from the cloud servers. While the adversary will not be able to access the file’s contents due to encryption, they can observe user interactions based on which files the user is requesting and receiving. To prevent even IP based analysis by cloud provider, the user might utilize multiple clouds for different groups s/he is managing. Moreover, friends can utilize proxies and anonymizer technologies such as Tor or I2P [20], in her/his communications with the cloud. Anonymizers would help users maintain their privacy by exchanging content without the adversary being able to link the usage to a specific user. Additionally, rather than relying on other anonymizers, users could potentially ask a random friend to perform cloud downloads on their behalf so that the cloud provider cannot detect who is accessing the content. Such random facilitators would protect a user’s activities from watchful cloud providers as there will be a larger number of IPs accessing content with same as facilitators. Note that, to download content, a friend just uses direct link to the file without logging into the cloud. While this approach would obscure user interactions, it would however add to the user overhead. The directories and file names stored in the cloud could also be randomized to further protect the users’ privacy from the cloud provider.

Another solution that can be used if the user does not even want to use cloud because of privacy concerns, the multimedia can be disseminated directly from peer to peer. One approach to achieve this is, sender can send the multimedia to all of the friends who requests the multimedia directly. However, this method adds a considerable bandwidth overhead on the sender. In order to reduce the overhead on
the sender, we can build a chain like system where the initial peer is the sender. Sender sends the multimedia to one of the online friends, upon receiving the file, this friend sends it to the next friend in the chain. While this approach helps to reduce the overhead on sender, it will introduce extra latency. It will add extra latency on the delivery of content to the peers towards the end of the chain. The latency and overhead can be optimized by building a multicast tree through online friends. In this case, the user sends the multimedia to a selected number of online friends, and these friends forward it to other set of online friends.
Chapter 5

Cloud as Storage

In this chapter, we compare the communication performance of the storage clouds to OSNs. As proposed decentralized system relies on storage clouds to share encrypted user data, we would like to see if storage clouds can perform data communication as efficient as OSNs that are optimized for user interaction. We focus on comparing the download and upload performance of popular social networks (i.e., Facebook and Google+) to cloud providers (i.e., CopyCloud, Dropbox, GoogleDrive, Mediafire, and OneDrive). These measurements could also help to determine which cloud providers would be more efficient in supporting POSN functionality.

5.1 3G/4G Measurements

We first carried out measurements with 3G/4G cellular networks to compare Dropbox and Facebook communication performance.\(^1\) Volunteers on the West Coast of the United States were asked to install and run an application that reported download measurements with various file sizes to Facebook and Dropbox in random order. Seven

\(^1\)These measurements were conducted with code developed by James Bridegum.
volunteers participated with different devices and provided measurements. Each volunteer did not complete the same number of experiments, due to the large amount of data consumption required to perform the measurements over cellular network.

Figure 5.1 presents the ratio of Dropbox to Facebook performance for downloading different files for the same device. There was a considerable performance variation in data transfers over cellular networks, and this is likely due to the instability in the cellular bandwidth rate. Therefore, the ratios are presented rather than the timing of each device. Each point in the figure indicates one pair of measurements for the device at the same location performed in random sequence. If the point has a value above one, this indicates that the Facebook is faster than the Dropbox. Overall, it was observed that Facebook can be faster up to 5.5x and that Dropbox can be up to
9x faster. The measurements showed that Facebook is faster for the majority of the time (up to twice for all but one device). This results indicate that the data transfers in Facebook is better optimized than Dropbox in delivery of content, especially for smaller data.

5.2 Wi-Fi Measurements

To compare the performance of the OSNs and cloud providers in detail, we performed measurements with various file sizes with different OSN and cloud providers over WiFi in a better controlled environment. The file sizes included 2 KB to 718 KB to simulate photos and 1 MB to 100 MB to simulate video sharing. Facebook compresses pictures during re-rendering and we were not able to upload and then download a picture larger than 718 KB. Hence, 1 MB and larger files are videos doubling in size whereas below 1 MB are pictures resized by Facebook. All of the measurements were performed during the morning or early afternoon to avoid the peak Internet traffic times.

We performed WiFi experiments on a Samsung Galaxy S4 phone at nine different locations around the Reno, Nevada, USA. The different locations had either a public Wi-Fi connection or a private one. The locations with free public Wi-Fi include: an airport, a bookstore, a cafe, a public library, and a mall. The private Wi-Fi locations were three different ISP providers at different residential locations and a university campus. These different locations were selected to simulate scenarios where people would commonly access social networks on their mobile devices.

A small file was downloaded and uploaded to each provider to remove any

\footnote{The details of measurements are presented in \cite{27} by Eric Klukovich who conducted the measurements.}
network latency (such as sleeping radio or DNS resolution) that would occur when the device first started a set of tests. Every file size had twelve measurements, where the minimum and maximum values were removed to exclude outliers and the ten remaining times were averaged. The download order of the different file sizes and providers were randomly shuffled for each set of tests in order to reduce any effects of caching and any bias towards downloading and uploading all the files to one provider.

Figures 5.2 and 5.3 display the throughput and time, respectively, performance of the measurement for all locations with different file sizes. Each point shows the average of 90 measurements (10 performed at 9 locations) for upload and download of various size files.

From Figure 5.2, we observe that as file size increases the throughput increases. Among cloud providers Mediafire has the lowest throughput in uploading files. CopyCloud, Dropbox, and OneDrive all had very similar download and upload performances across all file sizes. When uploading the files, Facebook has the best upload performance for files less than 100 KB, but is similar to other providers for larger files. The rest of the providers show similar performance in uploading any file size, except for Mediafire, which often lags behind others. Overall, the providers had a linear throughput increase for uploading the photos, but then stayed at a constant rate for files greater than 1 MB (i.e., videos). The constant rate can be correlated to reaching the maximum upload bandwidth provided by the ISP. From this measurement, we realize that the upload performance is very comparable for files greater than 100 KB for the OSN and cloud providers.

The download bandwidth for the files increased linearly with the size of the file up to 1 MB and then varied. The linear increase in this graph indicates that the server is not the bottleneck, but that the network and other factors affect the
Figure 5.2: Average throughput for different file sizes (log-log scale)

performance. Considering download throughput, GoogleDrive and Mediafire had the worst performance for files less than 200 KB, but then had similar performance with other providers for larger files. Facebook and Google+ showed very similar download performance except for the files larger than 1 MB (i.e., videos). The performance hit of Google+ is due to the limitation of their API where videos are downloaded as a stream rather than a single file causing an impact on the performance.
From Figure 5.3, we observe that the upload and download time for the photos (i.e., files less than 1 MB) increases slightly as the size of the photo became larger but for the videos there is a linear increase with the file size. Compared to other providers, GoogleDrive and Mediafire both had lower throughput and slower times when downloading any of the photos. The reason for this difference in performance is likely due to how the API for each provider requires two connections for one download,
one connection to query to get either the file ID for GoogleDrive or a query for
the download link in Mediafire, and a second connection to download the file. The
other providers can download files using only one connection by using the actual file
name to download the file. The measurements showed that Facebook had the best
upload timing performance, while Mediafire had the worst. Overall, while Facebook
seem to be better optimized to upload for small photos (less than 100 KB) and
videos (less than 4 MB), other providers had similar timing in uploading content.
When downloading content, however, all providers have similar performance except
MediaFire and GoogleDrive for small files (less than 200 KB), due to extra query to
download a file in their API.

When the results are examined independently, we notice there is a performance
increase (i.e., decrease in time with a throughput jump) when the file size goes from
the largest image (718.43 KB) to the smallest video file (1 MB). This performance
change is likely due to TCP slow start and congestion control scheme. The process
starts off by sending a few segments initially and waits for the handshake to occur.
Once the acknowledgement is received, then more packets are sent, by an exponential
amount, and the process repeats until the threshold is met. TCP takes a while to
achieve full steady state behavior and smaller sized files (less than 1 MB) may not
even reach this state. When the file size is about 1 MB, then the TCP connection
could have full bandwidth and outperform slightly smaller files.

Overall, compared to the cloud providers, Facebook and Google+ were the
most efficient for downloading photos in almost all the locations. This is most prob-
ably because of the optimization that the OSN providers have in place to deliver
content in an efficient manner. Facebook also showed the best performance when
uploading images compared to the other providers. Note that uploaded images were
rendered by Facebook earlier and new photos would have rendering overhead when uploaded to Facebook, which is excluded from our measurements.

When considering cloud providers, OneDrive was found to be the most efficient in terms of download and upload throughput and Mediafire tended to have the worst performance especially for images. Mediafire has the additional query overhead and lack of an Android API to access their services which causes the worst performance, whereas the other providers have an optimized Android API for their services. Dropbox and Mediafire showed the worst performance in many of the video tests. Google+ was found to have the slowest download times for the videos as well due to the limitations of their API.

Figure 5.4 presents the average time and throughput performance of OSN and cloud providers when all are combined. Overall, the OSNs are slightly better than the cloud providers in download performance for file sizes less than 16 MB and worse for larger ones. However, this difference, on average, is 82% higher for download and is only 36% higher for upload time. When considering one of the fastest clouds, OneDrive is only 38% slower than Facebook in downloading various size content. Overall, there was no significant performance difference when downloading and uploading a file to or from an OSN or a cloud provider.

5.3 Cloud Performance Analysis

Overall, the OSN providers had a slightly better communication performance when delivering small files, but the cloud providers had better performance for the larger files. GoogleDrive and Mediafire showed the slowest data transfer performance in part due to the querying architecture and limitations of their APIs. The communication
performance of the social networks and fastest cloud providers was found to be very comparable, even though their primary target of content delivery differs. For instance, OneDrive is only 38% slower than Facebook, in downloading any file on average. These results indicate that we can rely on the storage clouds to efficiently disseminate user data, especially with larger data files. Hence, in our design, we prioritize device-to-device interaction for short messages and wall posts, and rely on cloud to distribute...
larger multimedia content.

5.4 Potential Storage Usage

In the Facebook study presented in Chapter 3, we collected the detailed activity of 16 Facebook users and presented how active the friend circles of these user were. Ignoring friends that never appear to be online, the average number of comments by a friend in user’ circle is 16.66, status updates is 5.42, links is 3.80, pictures is 5.25, and videos is 0.73 for the 15 day duration. Based on the content generated by all the users and their friends within this period, we calculated the total number of posts a typical user would make in Table 5.1. These statistics would lead to 405 comments, 132 status updates, 92 links, 128 pictures, and 18 videos for a year, on average per user. Ignoring offline friends, the average photo and video size per friend for 15 days were 292 KB and 6.77 MB, respectively, leading to a total content of 172 MB for a year. From these measurements, we observe that the amount of potential storage need will not exceed the free storage for most of the cloud providers even with several years of data. Note that, while friend’s most recent content is cached on the application, only the user’s own content is stored in cloud. Additionally, very active users would need extra space either by purchasing from storage cloud provider or through a self hosted server that holds bulk of the data.
<table>
<thead>
<tr>
<th>User</th>
<th>15 Day Average</th>
<th>Yearly Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Num of Posts</td>
<td>Multimedia Size</td>
</tr>
<tr>
<td>User 1</td>
<td>21.91</td>
<td>8.47 MB</td>
</tr>
<tr>
<td>User 2</td>
<td>13.15</td>
<td>4.76 MB</td>
</tr>
<tr>
<td>User 3</td>
<td>8.61</td>
<td>4.70 MB</td>
</tr>
<tr>
<td>User 4</td>
<td>7.69</td>
<td>6.62 MB</td>
</tr>
<tr>
<td>User 5</td>
<td>10.39</td>
<td>5.55 MB</td>
</tr>
<tr>
<td>User 6</td>
<td>13.85</td>
<td>6.67 MB</td>
</tr>
<tr>
<td>User 7</td>
<td>12.08</td>
<td>7.86 MB</td>
</tr>
<tr>
<td>User 8</td>
<td>10.15</td>
<td>0.90 MB</td>
</tr>
<tr>
<td>User 9</td>
<td>12.46</td>
<td>4.46 MB</td>
</tr>
<tr>
<td>User 10</td>
<td>20.51</td>
<td>8.96 MB</td>
</tr>
<tr>
<td>User 11</td>
<td>19.51</td>
<td>10.45 MB</td>
</tr>
<tr>
<td>User 12</td>
<td>10.92</td>
<td>8.30 MB</td>
</tr>
<tr>
<td>User 13</td>
<td>19.76</td>
<td>10.96 MB</td>
</tr>
<tr>
<td>User 14</td>
<td>5.63</td>
<td>1.93 MB</td>
</tr>
<tr>
<td>User 15</td>
<td>15.87</td>
<td>10.91 MB</td>
</tr>
<tr>
<td>User 16</td>
<td>76.81</td>
<td>8.57 MB</td>
</tr>
<tr>
<td>Average</td>
<td>17.46</td>
<td>6.88 MB</td>
</tr>
</tbody>
</table>
Chapter 6

Access Control

In this chapter, we provide details of the access control tokens that ensure only intended friends are able to access user data, how to share data with a friend, and how to integrate third party apps into POSN platform. POSN provides data confidentiality and fine grained access control.

6.1 Access Control Tokens

POSN separates data into different files using different encryption keys to allow for fine-grain access control. As data is decentralized, fine grained access control can not be provided through centralized authorities as in [32] but rather is provided through dissemination of access tokens that contain unique URLs and keys. Figure 6.1 shows how the files are organized in the cloud and how they are linked.

The encryption keys are transferred between users directly where the keys are encrypted with the user’s public key, or are placed in different friend, group, or user files, where the friend is given specific access. As a result, the access control scheme
Figure 6.1: POSN access tokens for Alice
and encryption keys are hidden from other friends and adversaries who do not have access, guaranteeing the user’s privacy and anonymity from observers.

In order to provide security and privacy, data is encrypted before uploading into the cloud. Public key’s of users are exchanged at the time of friendship establishment and shared through the cloud. POSN keeps a wall for each group as most users form clusters of friends (see Chapter 7). Therefore, each group needs a key for the group wall, and these keys need to be exchanged securely.

In order to handle the access to the wall post that belongs to a specific group, the poster embeds a symmetric key into the file. Using this mechanism, the symmetric keys do not need to be exchanged separately but rather recovered through the earlier keys. Furthermore, as multimedia files might be posted to different groups, we encrypt multimedia files with individual symmetric keys, which are shared in the wall post for the group.

6.1.1 Data Owner

The data owner file (Alice User File in the Figure) is created for account holder and contains information that is specific for that user. The file is encrypted with its own symmetric key and holds the data for the owner’s public key, IP address, port number, online status. This information allows friends to communicate with the user when they are online. The user’s friend list is also stored to allow friends to find mutual friends and improve data dissemination (Chapter 7). When a user creates a new post, the user selects which group to share the post with. The updated wall file is then uploaded into the cloud, and any online friend is notified of the new post. If the post contains any multimedia, then a new symmetric key is generated to encrypt the multimedia file. The multimedia file is uploaded into the cloud and a direct download
link for the file is appended to the post with its key.

6.1.2 Friend Access

Friend files are created for each friend and are used to contain tokens for the groups the friend can access. Each friend file can only be modified by the owner, and is encrypted with the friend’s public key to guarantee only the friend specified can access the file. The file contains (i) a link to the data owner file so that the friend can have access to the data owner’s online information and friend list, and (ii) a list of groups the friend is in along with their access tokens such as group ID, group wall URL, version, associated symmetric key. When a user becomes online, different wall files from all friends need to be fetched and decrypted using the appropriate symmetric keys. The post ID is used to determine which new posts need to be added to the newsfeed, and if any multimedia content needs to be fetched from the friends’ clouds. If a user cannot decrypt a group wall file, then either the user’s friendship has been revoked and s/he no longer has access to user content or the key was updated and the new version needs to be fetched.

6.1.3 Group Wall Access

The group wall files hold the data for individual posts that the data owner uploads for the specific group. Each post has its own set of associated meta-data such as post ID, post date, content type, user ID, data. If the post is a textual status update or a link, then the content is embedded directly with the meta-data. Multimedia posts require additional fields to hold the multimedia link in the cloud and the appropriate key. The wall file also contains a link and key to the group’s archive, if any. Archiving
helps to keep active files small and also renew the group keys. Each group wall file is encrypted with the group’s symmetric key that is stored in the friend file.

6.1.4 Archiving

As the data owner uploads new content to the social network, the group wall files will continuously grow in size, and therefore it would become slower for the friends to process the new posts. Users typically access a small number of objects among a vast number of posts, with many users accessing only recently posted objects. The purpose of the group archive file is to reduce the overhead by storing older posts into a separate file and minimizing the size of the active group wall file. The archive has the same format as the group wall file, and holds the link and key to the previous versions of the archive file, if any. The archive is encrypted with the appropriate version of the symmetric key, which is stored in the group wall file for quick access.

The archive files create a chain of files from the more recent posts down to the oldest posts. The purpose of the chaining the files is twofold: to organize the post history into different time periods, and to make the friend revoking process more efficient (see Section 6.2). The archiving process can be done periodically every week, month, or year depending on how often the data owner posts new content. The archive process will also be invoked when the data owner wishes to revoke a user from one or multiple groups. Since reorganizing the archive can require additional processing, the archiving can be done when the user’s device is charging and connected to a Wi-Fi connection to minimize the impact on the overall performance of the device.
6.2 Granting and Revoking Access

There may be a time where a user wishes to change what a friend can access, or even wish to no longer be connected with a friend and remove them from their circle completely. Adding a friend to a new group is straight-forward, where the new group information and key is appended to the friend’s file. Since all of the group archives and individual posts are accessible through the group file, no additional overhead is required. On the other hand, revoking a friend’s access can be a challenge, due to changing of the keys and possibly re-encrypting the files to prevent the user from accessing the content. This re-encryption process would add significant overhead to the system, and, in POSN, this overhead is avoided by archiving current content.

POSN cuts off a revoked user by archiving all groups that the removed friend is in and not sharing the new group wall keys with the removed friend. This approach has the benefit of not having to re-encrypt the old data, which the removed friend had access anyway. The revoked friend will only be able to access any content that was created before the revoking process. This also adds the benefit that the revoked user will not know that they have been removed, and they will think their friend is no longer posting new content. If we want to completely block a user from even viewing the previously shared content, we could change the archived file’s link without notifying the ex-friend. Even though the ex-friend has the decryption key, s/he would not be able to access data unless s/he has a local copy. The group wall links and keys will no longer be valid as the walls would be at new locations.

One of the challenges of current platform is that, the method of establishing friendship between two users can be susceptible to man-in-the-middle attacks by their email or SMS providers. If the user who initializes the friend request via a malicious
email server, then the initial friend request email could be sent to the adversary. In turn, the adversary can then spoof an email and provide their information to obtain the initiating user’s information to access content that was never intended to be shared with them. A solution to this issue would be to use two-factor verification mechanisms.

6.2.1 Content Sharing

Uploading and sharing content, whether it's a status update, picture, video, or a link, is a fundamental aspect of online social networking. Figure 6.2 shows how a post can be created in the POSN app. When creating a new post, the user is required to select which friend group the post will be shared with. If the user is creating a new multimedia post, then he or she must either select a photo/video from the gallery or take a new picture/video from the camera. The multimedia file is then encrypted with a new symmetric key and uploaded into the cloud, where the direct download link is fetched. The multimedia key and the direct download link are embedded with the meta-data into the corresponding group wall. The appropriate group wall files are then fetched and decrypted, so the new post can be appended to the wall files. The wall files are then uploaded into the cloud, and any online friends will be notified of the new post, so they can add it to their walls. The status post requires similar steps, except the status text is directly embedded in the post with meta-data. Any online friend will get a notification that a new post was created and they can fetch the multimedia.
6.2.2 Content Access

Accessing friend content is another fundamental part of social networking, and it needs to be as efficient as possible. When a user becomes online, different wall files from potentially all of the friends need to be fetched and decrypted using the appropriate symmetric keys. The post ID is used to determine whether new posts need to be added to the newsfeed, and if any multimedia content needs to be fetched from the friend’s cloud. If multimedia content is needed, then the direct download link and the multimedia symmetric key in the wall are used to fetch and view the content. If a user cannot decrypt the group wall file, this indicates the previous wall has been archived and the new key version needs to be fetched from friend file.
6.3 Third Party Application Support

Limiting the features of OSNs with only communication typically reduces its role. OSNs were mostly used for communication at the beginning, but this trend has changed over the time. Today, OSNs support various types of 3rd party applications. For instance, games are one of the applications that people like to play and compete with their friends. POSN framework can support real time and offline game playing with friends as it will establish a connection between friends. Health related applications are potential applications that POSN would provide considerable incentive for people to adopt. There are myriad of upcoming devices and sensors that collect different type of health measurements. Sensor data of mobile devices (such as physical activity and heart rate) can be processed by applications to track the user’s health and shared with friends to compete/motivate. In general, personal motivation is very important in terms of physical exercise and weight loss. Health applications can communicate with these devices and disseminate this information to relevant people (such as, doctors, family, etc.) Another popular type of application would be the location based applications. Families can use tracking applications to know each other’s location. Similarly, people that travel with a tour, can trace each other as they explore new locations. Frequently visited locations (such as frequently visited restaurant) can also be disseminated by the application for recommendation purposes.

Applications can be developed as integrated into the POSN system. However, it is better to provide support for 3rd party application developers who often come up with a wide variety of ideas. Integrated application development requires considerable manpower and time, and lacks diversity. Providing support for 3rd party
application may increase the popularity of POSN framework among different communities. The advantage of POSN framework is that information created or gathered by the applications remains in the system as there is no central server. Utilizing POSN, applications that require sensitive information can be used by the people without privacy concerns (as long as they are vetted not to leak information).

An important challenge is how to allow 3rd party applications access data while keeping the data in the circle of trust. To allow third party application support without harming user privacy, each user can keep an App File for each authorized application in their cloud. This App file is encrypted with a key similar to other group walls. Applications will write the relevant data into the files without revealing it to the app developer. Moreover, friends who want to partner in specific app can exchange their App file location and keys so that they can collaborate/compete. Online applications can also exchange information over the socket connection that is created by the online clients.
Chapter 7

Data Dissemination

In this chapter, we analyze different aspects of efficient data dissemination in a decentralized OSN platform such as POSN.

7.1 Cloud-backend

The first issue we analyze is whether to transmit the data directly to online peers or whether to rely on the cloud to disseminate content. As multimedia can be very large, the multimedia could be distributed through cloud where the user lets her/his online friends know that there is a new post with a direct link to download data from the cloud. Otherwise, the user would send multiple copies of the file (i.e., as many as the number of online friends) directly to her/his friends. In measurement study in Chapter 3, we observe that two thirds of photos are less than 68 KB while none is greater than 900 KB. Likewise, two thirds of videos are less than 20.2 MB but exceptionally there are videos larger than 100 MB.

Figure 7.1 shows the sender bandwidth overhead for the 16 users for the mul-
timedia posts they generated during the measurement period. While x-axis shows the number of online friends when the user posted a multimedia, y-axis indicates the total data amount to send a copy of the file to each online friend directly or through cloud. It indicates that user overhead could considerably increase if the user directly sends the multimedia files to online friends. For example, to send 118.6 KB photo to 77 online friends directly, the user would need to send over 6 MB data. Similarly, to send 7.5 MB video to 9 online friends directly, the user would need to send over 67 MB data.

In decentralized OSNs, rather than sending multiple copies of the multimedia file, the user should upload a copy to the cloud/server and notify online friends with
a direct link to the file. This also allows friends to only fetch a multimedia content if they are interested in viewing them, i.e., as they are scrolling through their wall.

7.2 Community-based Data Distribution

The lack of a central server results in accessing several locations to gather the posts from friends’ clouds or PCs. In the worst case, the user has to check all of friends’ cloud or PC locations. This may introduce significant overhead as there could be hundreds of friends whose repositories need to be checked. The number of locations to be accessed depends on the number of friends and would result in considerable overhead for users with many friends.

As people form different communities in their social or online interactions [31], we could take advantage of such communities to locate new content. The number of connections that a client needs to aggregate the information from can be considerably reduced as friends typically form community clusters. Figure 7.2 presents the friend circle of measured users. We detected communities in each user circle using Louvain method [12]. We observe that majority of the friends form tightly connected communities, shown with different colors. For all users, we have their circles in a few communities that are tightly interconnected.

The community clustering can be taken advantage of when distributing data. Online friends can exchange new content to any common friends. As a result, the number of connections that a user needs to aggregate the information from will significantly decrease.
Picture 7.2: Friend’s inter-connections for all users
7.3 Getting Online

A major issue in decentralized OSN platforms is how to identify online friends to establish direct connection. When there is no central server or central data repository, one potentially needs to check every friend’s repository.

When a client (such as client A in Figure 7.3) becomes online, it will look in the friends’ cloud/server to see if they have a new post and learn their communication address (i.e., IP address and port number) if they are online. Green nodes indicate an online friend, blue indicates an offline friend, * indicates a PC backend, and links indicate common friendship. If the friend (such as client B) is online then the client can establish a direct connection and ask the friend about common friends (such as clients C, D, L, and M in the example). The online friend then provides its knowledge about the common friends. Once other online friends (such as D) are reported, then the newcomer can recursively query these friends. If a friend is not online through a mobile device, but has the PC/server connected (such as E), then the user can directly connect with the PC/server. The PC/server can take in requests from the user for data about any common friends, even if the user’s mobile app is offline. As friends with a PC/server backup can provide interaction capability compared to cloud backed ones, a user can first check with them to obtain initial information about current state of common friends. This lookup operation is carried out only when a user becomes online. Thereafter, rather than periodically checking for new wall posts at every friend’s cloud/server, post notifications should be pushed to online friends.
7.4 Initial Lookup Approaches

The initial lookup overhead can be considerably reduced by contacting online friends with most information about the rest of the friends. Hence, the order of friend lookup is important as finding online friends early on could eliminate queries to other friends. The approaches to query order include to rank friends by the (i) number of common friends, (ii) their expected online duration, or (iii) random. In order to implement the first two approaches, a user needs to keep and share the relevant information with friends.

To assess the connection savings with different approaches, we compare four different fetching methods: ideal, most online duration, most common friends, and random for our measured data. The ideal provides the best case scenario with an oracle that tells who is online and has most information. Note that, in this simulation, we do not consider friends with a PC/server backup as they should be the first to interact with.
Figure 7.4: Likelihood of finding friends online at a given time with CDF distribution
In our simulations we take several snapshots of online activity. To mimic real user activity, we assumed the user came online one minute before one of her/his friends actually became online in the Facebook measurements. Hence, we obtain a much larger number of data points than an individual’s online pattern.

In Figure 7.5, we provide the results of connection saving with different approaches for three of the user’s circles. The rest of users’ simulations are provided in Appendix A. In the figure, the x-axis indicates the percentage of online friends when the user becomes online and the y-axis indicates percentage of connections that could be eliminated. Each point indicates one simulation scenario. The $\mu$ in the figure indicates the average connection saving. Table 7.1 presents the average of five-tuple summary statistic for 16 of the users, while the details are provided in Appendix A.

In POSN system, we take the advantage of online friends, to fetch and disseminate data. From the Facebook measurement study, we see that on average 12% of friends are always online. Figure 7.4 presents the likelihood of observing a given percentage of friends for 16 Facebook users, along with CDF distribution in Figure ???. The x-axis shows the percentage of friends that were online and the y-axis shows the likelihood of that number of friends being online. Individual figures for each user, that present the likelihood of observing a given percentage of friends for 16 Facebook users (along with CDF of the distribution), can be found in Appendix A. An important thing to note is, these friends are not necessarily the same each time. We observe

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>1st quartile</th>
<th>median</th>
<th>3rd quartile</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal</td>
<td>19.3</td>
<td>59.6</td>
<td>63.9</td>
<td>67.7</td>
<td>79.9</td>
</tr>
<tr>
<td>Most common</td>
<td>8.5</td>
<td>40.9</td>
<td>47.6</td>
<td>52.2</td>
<td>67.1</td>
</tr>
<tr>
<td>Most online</td>
<td>15.4</td>
<td>46.7</td>
<td>51.9</td>
<td>56.8</td>
<td>67.2</td>
</tr>
<tr>
<td>Random</td>
<td>5.9</td>
<td>33.5</td>
<td>38.8</td>
<td>43.7</td>
<td>63.4</td>
</tr>
</tbody>
</table>
Figure 7.5: Help from online friends
that the users who have fewer friends, have a higher likelihood of having more friends online for a longer period of time. Conversely, the users with more friends have a lower likelihood that a larger portion of their friends will be online for a long duration. For example, it is more likely for a user with 10 friends to see half of her/his friends online for a long time compared to a user with 500 friends.

These results indicate that as there are more friends a user can achieve greater saving by utilizing online friends. This is encouraging as the overhead for a larger circle would have been higher. Additionally, the simple heuristics of most online and most common methods achieve close to ideal savings, i.e., if an oracle told who is online at the moment. Even a random ordering (i.e., no heuristic use for ordering of lookups) can reap considerable connection savings by simply asking online friends for their knowledge of common friends.

7.5 Potential Initial Data

Another issue is the amount of data a user potentially needs to download when s/he becomes online. Figure 7.6 shows the amount of multimedia and non-multimedia shared by the friends of Facebook users that we have monitored. Based on this data, we calculated the size and number of posts shared by the friends of a user, when s/he is offline.

Figures 7.7-7.9 present the posts (i.e., video, photo, link, status, and comment) a user could download when s/he becomes online for three of the users while the rest of users’ simulations are provided in Appendix A. Similar to the previous simulation, we assume the friends’ online pattern as the user’s online duration. For instance, for User 1, we assume an online pattern that mimics each of the 635 friends separately.
Figure 7.6: Average number of posts by circles
Each point indicates one possible scenario considering the time since last login and the number of friends’ posts that has accumulated since. Hence, there are considerably more data points than an individual user’s. Note that the figures do not show the logins that had no post in the circle but the averages consider them.

For some users we observe two clusters due to the difference between photo and video sizes. A video post in a smaller friend circle becomes more obvious while a large circle has a large number of photos that accumulate between logins.

As the time between logins increase, the amount of data the user could download from her/his friends’ walls increases. In extreme cases, we observe there is about 1 GB of data when the user has not logged in for about a day and friends have posted large videos. Hence, rather than auto-downloading all multimedia content, application should parse most recent posts and wait for the user to fetch subsequent content as s/he is scrolling through the posts (especially for large videos). Table 7.2 and 7.3 tabulate the five-tuple statistic summary. As already mentioned in previous section, to mimic real user activity, we assumed the user came online one minute before one of her/his friends actually became online in the Facebook measurements. Hence, we obtain a much larger number of data points than an individual’s online pattern. In the tables, average inter-arrival times shows the average duration (in minute) the user goes offline that is calculated based on the online pattern of friends of the user. We calculated and presented the statistics of the incoming number of posts and data volume for each user, while the user is offline. The results indicate that as the time between logins increase, the amount of posts increase. Similarly, as a user has a larger circle, the amount of posts they could fetch increases.
Figure 7.7: Posts to download when a user becomes online (log-log scale)
Figure 7.8: Posts to download when a user becomes online (log-log scale) (cont.)
Figure 7.9: Posts to download when a user becomes online (log-log scale) (cont.)
Table 7.2: Posts to download when becoming online

<table>
<thead>
<tr>
<th></th>
<th>Average inter-arrival</th>
<th>Number of posts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>min</td>
</tr>
<tr>
<td>User 1</td>
<td>177</td>
<td>1</td>
</tr>
<tr>
<td>User 2</td>
<td>236</td>
<td>1</td>
</tr>
<tr>
<td>User 3</td>
<td>224</td>
<td>1</td>
</tr>
<tr>
<td>User 4</td>
<td>214</td>
<td>1</td>
</tr>
<tr>
<td>User 5</td>
<td>234</td>
<td>1</td>
</tr>
<tr>
<td>User 6</td>
<td>239</td>
<td>1</td>
</tr>
<tr>
<td>User 7</td>
<td>220</td>
<td>1</td>
</tr>
<tr>
<td>User 8</td>
<td>232</td>
<td>1</td>
</tr>
<tr>
<td>User 9</td>
<td>238</td>
<td>1</td>
</tr>
<tr>
<td>User 10</td>
<td>263</td>
<td>1</td>
</tr>
<tr>
<td>User 11</td>
<td>241</td>
<td>1</td>
</tr>
<tr>
<td>User 12</td>
<td>283</td>
<td>1</td>
</tr>
<tr>
<td>User 13</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>User 14</td>
<td>348</td>
<td>1</td>
</tr>
<tr>
<td>User 15</td>
<td>224</td>
<td>1</td>
</tr>
<tr>
<td>User 16</td>
<td>273</td>
<td>0</td>
</tr>
<tr>
<td>Average</td>
<td>244</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 7.3: Data to download when becoming online

<table>
<thead>
<tr>
<th>User</th>
<th>Average inter-arrival</th>
<th>min</th>
<th>1st quartile</th>
<th>median</th>
<th>3rd quartile</th>
<th>max</th>
<th>quartile</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>177</td>
<td>0.00</td>
<td>1.67</td>
<td>27.29</td>
<td>197.81</td>
<td>1005.60</td>
<td></td>
</tr>
<tr>
<td>User 2</td>
<td>236</td>
<td>0.00</td>
<td>1.99</td>
<td>10.49</td>
<td>56.14</td>
<td>896.45</td>
<td></td>
</tr>
<tr>
<td>User 3</td>
<td>224</td>
<td>0.00</td>
<td>1.13</td>
<td>6.30</td>
<td>53.37</td>
<td>845.55</td>
<td></td>
</tr>
<tr>
<td>User 4</td>
<td>214</td>
<td>0.00</td>
<td>1.08</td>
<td>8.42</td>
<td>82.19</td>
<td>1320.79</td>
<td></td>
</tr>
<tr>
<td>User 5</td>
<td>234</td>
<td>0.00</td>
<td>1.59</td>
<td>9.72</td>
<td>62.20</td>
<td>859.83</td>
<td></td>
</tr>
<tr>
<td>User 6</td>
<td>239</td>
<td>0.00</td>
<td>0.72</td>
<td>5.58</td>
<td>40.05</td>
<td>473.87</td>
<td></td>
</tr>
<tr>
<td>User 7</td>
<td>220</td>
<td>0.00</td>
<td>0.53</td>
<td>2.29</td>
<td>11.36</td>
<td>86.42</td>
<td></td>
</tr>
<tr>
<td>User 8</td>
<td>232</td>
<td>0.00</td>
<td>0.76</td>
<td>5.63</td>
<td>43.42</td>
<td>488.98</td>
<td></td>
</tr>
<tr>
<td>User 9</td>
<td>238</td>
<td>0.00</td>
<td>0.37</td>
<td>3.98</td>
<td>39.09</td>
<td>351.95</td>
<td></td>
</tr>
<tr>
<td>User 10</td>
<td>263</td>
<td>0.00</td>
<td>1.92</td>
<td>10.00</td>
<td>85.63</td>
<td>639.65</td>
<td></td>
</tr>
<tr>
<td>User 11</td>
<td>241</td>
<td>0.00</td>
<td>0.45</td>
<td>3.61</td>
<td>34.75</td>
<td>621.69</td>
<td></td>
</tr>
<tr>
<td>User 12</td>
<td>283</td>
<td>0.00</td>
<td>0.61</td>
<td>4.09</td>
<td>15.84</td>
<td>489.59</td>
<td></td>
</tr>
<tr>
<td>User 13</td>
<td>256</td>
<td>0.00</td>
<td>0.39</td>
<td>3.95</td>
<td>24.85</td>
<td>340.59</td>
<td></td>
</tr>
<tr>
<td>User 14</td>
<td>348</td>
<td>0.00</td>
<td>0.04</td>
<td>0.21</td>
<td>0.83</td>
<td>69.75</td>
<td></td>
</tr>
<tr>
<td>User 15</td>
<td>224</td>
<td>0.00</td>
<td>0.10</td>
<td>1.27</td>
<td>5.01</td>
<td>31.26</td>
<td></td>
</tr>
<tr>
<td>User 16</td>
<td>273</td>
<td>0.00</td>
<td>0.35</td>
<td>2.53</td>
<td>18.97</td>
<td>244.02</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>244</td>
<td>0.00</td>
<td>0.86</td>
<td>6.59</td>
<td>48.22</td>
<td>547.87</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 8

User Interactions

In this chapter, we analyze how to implement timely data exchange in support of interactions between users. In particular, we propose user managed content propagation and delegated content propagation to disseminate user comments and such objects in a timely and efficient manner.

8.1 Commenting

Commenting is one of the most popular feature of OSNs as it provides interaction among friends and hence it should be incorporated into the OSN frameworks. However, majority of the decentralized OSN platforms ignore this feature as it is very challenging to provide organized commenting in a decentralized architecture. Commenting requires to organize many small objects and deliver them in a timely manner. Some decentralized OSNs use central repositories, such as Distributed Hash Table, to store such data. Although these dedicated repositories encrypt messages to keep them confidential, this approach still allows third parties to monitor the interactions
among users.

In POSN, commenting is achieved by exchanging comments among online friends and integrating them into wall posts as metadata when user is online or temporary files when post owner is offline. Figure 8.1 presents a general overview of how comments for a post by Alice are aggregated to form comment threads. Propagation of comments can be either managed by the user when s/he is online, or by a delegation process as explained below.
8.2 Comments as Objects versus Files

Comments can either be embedded directly within meta-data of the post in the cloud or held in files, where each comment has its own file. Embedding the comments brings the benefit of the user’s friends only needing to download a single file to get the most recent comments. A drawback is that the wall file size can grow large as there can be many comments for a single post. If the comments are stored in individual files, the user’s friends only need to fetch the comment file that they need to view and they can directly append their comments to the file via a unique URL. However, there can be a large amount of comment files since each post could have its own comment file. If there are too many files within a directory, the performance can decrease significantly, which in turn impacts the dissemination efficiency.

In order to determine which design choice is best suited for POSN, we implemented both methods and measured the data transfer performance. The measurements start with a wall post and gradually increases up to 5,000 uploads and 500 downloads. Each wall post contains varying number of comments and the number of characters per comment. The values were estimated from a power-law distribution based on the collected Facebook measurements.

Figure 8.2 presents the comparison for downloading and uploading comments embedded in the wall or as separate files. The comment links method has a very poor performance as it requires all the comment files to be downloaded sequentially, even with pipelining. As each comment is within a separate file the download performance gets worse as several small files need to be fetched from the cloud. The embedded comments only requires a single wall file to be downloaded to get the most recent comments and posts. When considering uploading, both methods show a linear
Figure 8.2: Performance comparison of embedded comments vs. comment file links growth, where the embedded comments outperform the comment links until about the 650th post range. After that point, the embedded posts has worse performance as the number of posts in the wall file grows. Since the performance for the embedded comments can grow to be significantly worse, there should be a threshold of about 500 posts in a wall to guarantee minimal upload overhead. The older posts should
be moved to archives as discussed in Chapter 6. Archiving also keeps the active wall file sizes small. Overall, embedding comments within a post has much better performance compared to using comment links with each post especially by limiting wall file size.

8.3 User Managed Propagation

In user managed propagation, post owner is responsible for creating and managing the comment files and such dynamic objects. Depending on online/offline status of the post owner, different actions can be taken as shown in Figure 8.3. Assume Alice is the post owner and Bob and Eric are her online friends. Below are the three cases.

![Figure 8.3: User managed propagation](image)

(i) When a friend makes a comment on a data owner’s post, the comment can be directly sent to the data owner if s/he is online. When Alice is online, she gets the comments from her friends and appends them into her post and notifies other online friends of the update.
(ii) If Alice is offline, her friends (e.g., Eric) can directly append the comment to a shared temporary file in Alice’s repository if the friends share the same cloud provider with Alice or Alice’s cloud allows write access through unique URLs. Alice’s other friends can check the temporary file to see if any comment is available for any of her posts.

(iii) When her friends (e.g., Bob) cannot directly append the comment to a file in Alice’s repository, they can either hold it until Alice becomes online or store the comment into a temporary file in their own cloud. In the later case, Alice could point to such temporary files of her friends so that others can fetch them while she is offline. This, however, could potentially incur a large number of lookups.

Alice would aggregate comments from such temporal files and append them to the original post when she becomes online. Hence her friends, could have the option of not querying her friends for comments but rather get them when she updates the post.

To handle these distributed commenting files, the post owner needs to have a file that holds the friend’s information where a link to each friends’ comment file is at (such as Friend List in Figure 6.1). Gathering comments from each friends’ cloud can introduce considerable overhead to the system. To minimize this overhead, we implement caching schemes as described in Chapter 7. When making a comment to a post, the user sends this comment to all (common) online friends who can further propagate to other online friends.
8.4 Delegated Propagation

Distributing the comments in a timely manner when the user is offline is difficult and incurs considerable overhead to obtain them from potentially hundreds of friends. In order to address this challenge, we can utilize delegation mechanisms where the data owner selects one of the online friends as a delegate and indicate this delegation in her/his status. When the user is offline, the comments are delegated to the chosen friend, and s/he collects and disseminates the comments on behalf of the offline user. In Figure 8.4, before going offline, the data owner Alice delegates her comments to Bob. If Bob has to go offline as well, he further delegates Alice’s comments to Chris (along with his), and when Chris needs to go offline he further delegates to David. If Bob becomes online while David is handling his comments, Bob can revoke this delegation and fetch the data from David for himself and Alice.

Such delegation approach could be managed per wall group to prevent non-group member friends detect posts not shared with them. It is also important to select right delegate to minimize overhead and prevent long delegation chains. The user could have four delegations options:

(i) a random friend,

(ii) friend that has the highest number of common friends,

(iii) friend who has recently been online the most, or
(iv) friend with the most disjoint online timing.

Each of these methods can use the prior ones as tie breaker. Using delegation, we can avoid costly lookup operations for any comment in a user’s circle.
Chapter 9

Conclusion

OSNs have gained a great importance in our daily life as people interact with their friends and acquaintances and even strangers through OSNs. As popular OSNs use a centralized architecture and personal data is controlled by the provider, privacy concerns have emerged. In this dissertation, we introduced a privacy preserving decentralized OSN (POSN) platform using mobile devices and (free) storage clouds. With the recent developments in smart mobile devices, OSN usage on mobile devices have considerably increased which would be valuable for a social network platform. The proposed platform utilizes mobile devices in a peer-to-peer architecture, along with the cloud storage as a highly available backup. It gives the user full control over their data, as content is only shared with the intended peers directly by the user.

To balance between technical efficiency of the solution versus social restrictions that exist in our daily life, we performed a measurement study with volunteers that allowed us to monitor the activity in their friend circles. We measured posting and online patterns in user’s circles, and analyzed how data is disseminated. Additionally, we measured the performance of popular social networks and cloud providers to com-
pare their communication efficiency. Overall, the OSN providers had a slightly better communication performance when delivering small files, but the cloud providers had better performance for the larger files. The communication performance of the social networks and cloud providers was found to be comparable, even though their primary target of content delivery differs.

In order to ensure data confidentiality, only encrypted data is transferred to the cloud. Additionally, access control is directly managed by the user via the cloud API and access tokens. The data stored in the cloud is separated into different files and encrypted with different keys to allow a fine grained access control. The encryption keys are transferred between users directly where the keys are encrypted with the user’s public key, or are placed in different friend, group, or user files, where the friend is given specific access. As a result, the access control scheme and encryption keys are hidden from other friends and adversaries who do not have access, guaranteeing the user’s privacy and anonymity from observers. It also removes centralized identity management that exists in some decentralized architectures.

Dissemination of data in a decentralized platform is one of the biggest challenges. POSN design focuses on the community of individuals and tries to optimize the data dissemination through device-to-device interaction and encrypted cloud storage. With the help of online friends and existing communities in the user circles, we showed how the data can be disseminated efficiently in a decentralized OSN environment. Another important feature of OSNs is commenting, which allows support interactions between users further. In this work, we also analyzed how to implement timely data exchange in support of interactions between users. In particular, we proposed user managed content propagation and delegated content propagation to disseminate user comments and such small objects in a timely and efficient manner.
In conclusion, POSN, incorporates popular OSN functionalities from commenting to instant messaging in a decentralized architecture among mobile devices with the help of cloud storage to store and disseminate data, which enables the friends access data even if the data owner is offline. POSN prototype is available at https://github.com/posn/POSN-app.
Chapter 10

Future Work

One of the features of the OSNs is the efficient search for objects. Searching is not a straightforward process in decentralized systems since there is no central database in the system. In POSN, the cloud is used only for storage purposes and friends content is scattered across several locations. In order to search for a content among friends, the wall files from all friends should be downloaded to a client to be searched through. Such a scheme is very inefficient since the number of content belonging to a user can be very high. Furthermore, this search is not limited only with the owner’s content, as it should include the content of each friend. It is almost prohibitive to download all these content, decrypt them, and search for desired content. We have proposed a two level indexing mechanism which needs to be further optimized and implemented.

Moreover, users with low resources (such as bandwidth cost or battery level) can benefit from computing and communication capacities of their online friends (such as being on Wi-Fi or having near full battery). Users would be inclined to help out their friends with phone-to-phone capacity sharing in POSN. We will investigate how to efficiently and fairly perform such phone-to-phone capacity sharing.
We currently have a developed POSN app publicly available, where we have some of the features implemented such as friendship establishment, posting a new content. We would like to further implement efficient data dissemination and comment propagation to POSN app.
Appendix A

Measurement Results
A.1 Hourly Content Posting Pattern

Figure A.1: Hourly number of posts by circles of User 1
Figure A.2: Hourly number of posts by circles of User 2
Figure A.3: Hourly number of posts by circles of User 3
Figure A.4: Hourly number of posts by circles of User 4
Figure A.5: Hourly number of posts by circles of User 5
Figure A.6: Hourly number of posts by circles of User 6
Figure A.7: Hourly number of posts by circles of User 7
Figure A.8: Hourly number of posts by circles of User 8
Figure A.9: Hourly number of posts by circles of User 9
Figure A.10: Hourly number of posts by circles of User 10
Figure A.11: Hourly number of posts by circles of User 11
Figure A.12: Hourly number of posts by circles of User 12
Figure A.13: Hourly number of posts by circles of User 13
Figure A.14: Hourly number of posts by circles of User 14
Figure A.15: Hourly number of posts by circles of User 15
Figure A.16: Hourly number of posts by circles of User 16
A.2 Multimedia Posting Pattern

![Graph showing multimedia posting pattern for User 1 and User 2.](image)

Figure A.17: Photo and video posting pattern of the circles of User 1 and 2 (log-log scale)
Figure A.18: Photo and video posting pattern of the circles of User 3 and 4 (log-log scale)
Figure A.19: Photo and video posting pattern of the circles of User 5 and 6 (log-log scale)
Figure A.20: Photo and video posting pattern of the circles of User 7 and 8 (log-log scale)
Figure A.21: Photo and video posting pattern of the circles of User 9 and 10 (log-log scale)
Figure A.22: Photo and video posting pattern of the circles of User 11 and 12 (log-log scale)
Figure A.23: Photo and video posting pattern of the circles of User 13 and 14 (log-log scale)
Figure A.24: Photo and video posting pattern of the circles of User 15 and 16 (log-log scale)
A.3 Online Pattern

Figure A.25: Number of online friends (in %) of users (minutely)
Figure A.26: Number of online friends (in %) of users (minutely) (cont.)
Figure A.27: Number of online friends (in %) of users (minutely) (cont.)
Figure A.28: Number of online friends (in %) of users (minutely) (cont.)
A.4 Online Rank

Figure A.29: Online duration pattern of the circles of User 1 and 2 (and its CDF distribution)
Figure A.30: Online duration pattern of the circles of User 3 and 4 (and its CDF distribution)
Figure A.31: Online duration pattern of the circles of User 5 and 6 (and its CDF distribution)
Figure A.32: Online duration pattern of the circles of User 7 and 8 (and its CDF distribution)
Figure A.33: Online duration pattern of the circles of User 9 and 10 (and its CDF distribution)
Figure A.34: Online duration pattern of the circles of User 11 and 12 (and its CDF distribution)
Figure A.35: Online duration pattern of the circles of User 13 and 14 (and its CDF distribution)
Figure A.36: Online duration pattern of the circles of User 15 and 16 (and its CDF distribution)
Figure A.37: Likelihood of finding friends online at a given time with CDF distribution for User 1
Figure A.38: Likelihood of finding friends online at a given time with CDF distribution for User 2
Figure A.39: Likelihood of finding friends online at a given time with CDF distribution for User 3
Figure A.40: Likelihood of finding friends online at a given time with CDF distribution for User 4
Figure A.41: Likelihood of finding friends online at a given time with CDF distribution for User 5
Figure A.42: Likelihood of finding friends online at a given time with CDF distribution for User 6
Figure A.43: Likelihood of finding friends online at a given time with CDF distribution for User 7
Figure A.44: Likelihood of finding friends online at a given time with CDF distribution for User 8
Figure A.45: Likelihood of finding friends online at a given time with CDF distribution for User 9
Figure A.46: Likelihood of finding friends online at a given time with CDF distribution for User 10
Figure A.47: Likelihood of finding friends online at a given time with CDF distribution for User 11
Figure A.48: Likelihood of finding friends online at a given time with CDF distribution for User 12
Figure A.49: Likelihood of finding friends online at a given time with CDF distribution for User 13
Figure A.50: Likelihood of finding friends online at a given time with CDF distribution for User 14
Figure A.51: Likelihood of finding friends online at a given time with CDF distribution for User 15
Figure A.52: Likelihood of finding friends online at a given time with CDF distribution for User 16
A.6 Online Help

Figure A.53: Connection saving when getting help from online friends for User 1
Figure A.54: Connection saving when getting help from online friends for User 2
Figure A.55: Connection saving when getting help from online friends for User 3
Figure A.56: Connection saving when getting help from online friends for User 4
Figure A.57: Connection saving when getting help from online friends for User 5
Figure A.58: Connection saving when getting help from online friends for User 6
Figure A.59: Connection saving when getting help from online friends for User 7
Figure A.60: Connection saving when getting help from online friends for User 8
Figure A.61: Connection saving when getting help from online friends for User 9
Figure A.62: Connection saving when getting help from online friends for User 10
Figure A.63: Connection saving when getting help from online friends for User 11
Figure A.64: Connection saving when getting help from online friends for User 12
Figure A.65: Connection saving when getting help from online friends for User 13
Figure A.66: Connection saving when getting help from online friends for User 14
Figure A.67: Connection saving when getting help from online friends for User 15
Figure A.68: Connection saving when getting help from online friends for User 16
A.7 Connection Saving

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>1st quartile</th>
<th>median</th>
<th>3rd quartile</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>53.7</td>
<td>74.2</td>
<td>78.0</td>
<td>80.1</td>
<td>83.2</td>
</tr>
<tr>
<td>User 2</td>
<td>4.6</td>
<td>74.3</td>
<td>79.4</td>
<td>82.1</td>
<td>89.6</td>
</tr>
<tr>
<td>User 3</td>
<td>44.0</td>
<td>68.4</td>
<td>72.2</td>
<td>75.8</td>
<td>81.6</td>
</tr>
<tr>
<td>User 4</td>
<td>48.8</td>
<td>73.0</td>
<td>76.7</td>
<td>78.8</td>
<td>83.8</td>
</tr>
<tr>
<td>User 5</td>
<td>8.3</td>
<td>55.5</td>
<td>59.7</td>
<td>63.1</td>
<td>72.5</td>
</tr>
<tr>
<td>User 6</td>
<td>4.9</td>
<td>52.2</td>
<td>56.5</td>
<td>60.4</td>
<td>73.1</td>
</tr>
<tr>
<td>User 7</td>
<td>12.5</td>
<td>67.5</td>
<td>71.8</td>
<td>75.4</td>
<td>90.4</td>
</tr>
<tr>
<td>Ideal User 8</td>
<td>13.7</td>
<td>58.1</td>
<td>62.8</td>
<td>67.6</td>
<td>81.8</td>
</tr>
<tr>
<td>User 9</td>
<td>5.7</td>
<td>56.3</td>
<td>60.1</td>
<td>62.4</td>
<td>69.9</td>
</tr>
<tr>
<td>User 10</td>
<td>4.7</td>
<td>28.1</td>
<td>33.5</td>
<td>39.8</td>
<td>77.4</td>
</tr>
<tr>
<td>User 11</td>
<td>12.9</td>
<td>86.2</td>
<td>87.9</td>
<td>88.7</td>
<td>93.8</td>
</tr>
<tr>
<td>User 12</td>
<td>3.0</td>
<td>48.1</td>
<td>55.0</td>
<td>61.1</td>
<td>80.0</td>
</tr>
<tr>
<td>User 13</td>
<td>13.1</td>
<td>42.7</td>
<td>46.8</td>
<td>50.8</td>
<td>64.4</td>
</tr>
<tr>
<td>User 14</td>
<td>30.1</td>
<td>69.8</td>
<td>72.7</td>
<td>75.7</td>
<td>84.5</td>
</tr>
<tr>
<td>Most common  User 8</td>
<td>6.2</td>
<td>33.1</td>
<td>39.7</td>
<td>45.7</td>
<td>66.1</td>
</tr>
<tr>
<td>User 9</td>
<td>1.9</td>
<td>35.3</td>
<td>39.1</td>
<td>42.1</td>
<td>54.8</td>
</tr>
<tr>
<td>User 10</td>
<td>2.3</td>
<td>10.9</td>
<td>17.1</td>
<td>22.6</td>
<td>72.6</td>
</tr>
<tr>
<td>User 11</td>
<td>3.7</td>
<td>62.9</td>
<td>79.6</td>
<td>82.1</td>
<td>89.6</td>
</tr>
<tr>
<td>User 12</td>
<td>0.4</td>
<td>29.1</td>
<td>37.2</td>
<td>43.0</td>
<td>65.0</td>
</tr>
<tr>
<td>User 13</td>
<td>7.7</td>
<td>29.3</td>
<td>32.0</td>
<td>37.4</td>
<td>50.9</td>
</tr>
<tr>
<td>User 14</td>
<td>4.4</td>
<td>62.7</td>
<td>68.6</td>
<td>74.5</td>
<td>81.9</td>
</tr>
<tr>
<td>User 15</td>
<td>8.4</td>
<td>37.5</td>
<td>50.0</td>
<td>54.2</td>
<td>58.4</td>
</tr>
<tr>
<td>User 16</td>
<td>16.6</td>
<td>27.1</td>
<td>37.5</td>
<td>43.7</td>
<td>64.5</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>1st quartile</td>
<td>median</td>
<td>3rd quartile</td>
<td>max</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>--------------</td>
<td>--------</td>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td><strong>User 1</strong></td>
<td>35.9</td>
<td>53.9</td>
<td>58.4</td>
<td>62.6</td>
<td>67.9</td>
</tr>
<tr>
<td><strong>User 2</strong></td>
<td>4.3</td>
<td>58.9</td>
<td>65.0</td>
<td>68.9</td>
<td>78.6</td>
</tr>
<tr>
<td><strong>User 3</strong></td>
<td>31.3</td>
<td>52.3</td>
<td>57.0</td>
<td>59.9</td>
<td>68.0</td>
</tr>
<tr>
<td><strong>User 4</strong></td>
<td>34.7</td>
<td>55.2</td>
<td>59.2</td>
<td>62.2</td>
<td>69.8</td>
</tr>
<tr>
<td><strong>User 5</strong></td>
<td>0.8</td>
<td>38.1</td>
<td>42.0</td>
<td>45.1</td>
<td>53.8</td>
</tr>
<tr>
<td><strong>User 6</strong></td>
<td>3.9</td>
<td>40.4</td>
<td>44.3</td>
<td>48.3</td>
<td>59.0</td>
</tr>
<tr>
<td><strong>User 7</strong></td>
<td>20.0</td>
<td>66.4</td>
<td>69.2</td>
<td>72.1</td>
<td>77.8</td>
</tr>
<tr>
<td><strong>Most online</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>User 8</strong></td>
<td>9.9</td>
<td>40.1</td>
<td>44.3</td>
<td>47.9</td>
<td>63.9</td>
</tr>
<tr>
<td><strong>User 9</strong></td>
<td>3.7</td>
<td>42.1</td>
<td>45.1</td>
<td>47.3</td>
<td>55.6</td>
</tr>
<tr>
<td><strong>User 10</strong></td>
<td>2.7</td>
<td>22.6</td>
<td>27.3</td>
<td>32.0</td>
<td>50.0</td>
</tr>
<tr>
<td><strong>User 11</strong></td>
<td>10.3</td>
<td>56.3</td>
<td>66.4</td>
<td>89.7</td>
<td>91.3</td>
</tr>
<tr>
<td><strong>User 12</strong></td>
<td>3.7</td>
<td>38.0</td>
<td>44.1</td>
<td>49.0</td>
<td>63.2</td>
</tr>
<tr>
<td><strong>User 13</strong></td>
<td>11.0</td>
<td>37.3</td>
<td>42.7</td>
<td>46.1</td>
<td>59.0</td>
</tr>
<tr>
<td><strong>User 14</strong></td>
<td>28.6</td>
<td>59.5</td>
<td>69.8</td>
<td>74.2</td>
<td>83.1</td>
</tr>
<tr>
<td><strong>User 15</strong></td>
<td>20.6</td>
<td>45.6</td>
<td>49.8</td>
<td>54.1</td>
<td>66.5</td>
</tr>
<tr>
<td><strong>User 16</strong></td>
<td>25.6</td>
<td>40.2</td>
<td>46.5</td>
<td>50.6</td>
<td>67.4</td>
</tr>
<tr>
<td><strong>Random</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>User 8</strong></td>
<td>0.4</td>
<td>29.2</td>
<td>33.65</td>
<td>37.5</td>
<td>55.1</td>
</tr>
<tr>
<td><strong>User 9</strong></td>
<td>3.5</td>
<td>58.5</td>
<td>62.8</td>
<td>66.4</td>
<td>81.1</td>
</tr>
<tr>
<td><strong>User 10</strong></td>
<td>0.4</td>
<td>24.9</td>
<td>30.3</td>
<td>35.3</td>
<td>56.1</td>
</tr>
<tr>
<td><strong>User 11</strong></td>
<td>0.7</td>
<td>31.5</td>
<td>36.0</td>
<td>39.8</td>
<td>54.1</td>
</tr>
<tr>
<td><strong>User 12</strong></td>
<td>1.9</td>
<td>14.8</td>
<td>18.7</td>
<td>23.4</td>
<td>67.1</td>
</tr>
<tr>
<td><strong>User 13</strong></td>
<td>1.7</td>
<td>53.4</td>
<td>59.4</td>
<td>64.6</td>
<td>90.0</td>
</tr>
<tr>
<td><strong>User 14</strong></td>
<td>0.7</td>
<td>24.0</td>
<td>31.1</td>
<td>37.1</td>
<td>61.0</td>
</tr>
<tr>
<td><strong>User 15</strong></td>
<td>3.7</td>
<td>18.5</td>
<td>23.9</td>
<td>28.0</td>
<td>45.5</td>
</tr>
<tr>
<td><strong>User 16</strong></td>
<td>2.5</td>
<td>16.0</td>
<td>22.2</td>
<td>28.5</td>
<td>51.5</td>
</tr>
</tbody>
</table>
A.8 Posts to Download

![Graph showing posts to download when User 1 becomes online](image)

Figure A.69: Posts to download when User 1 becomes online (log-log scale)
Figure A.70: Posts to download when User 2 becomes online (log-log scale)
<table>
<thead>
<tr>
<th>Number of Posts</th>
<th>Total Size (MB)</th>
<th>Interval (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>0.1</td>
<td>100</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td>1000</td>
</tr>
</tbody>
</table>

Figure A.71: Posts to download when User 3 becomes online (log-log scale)
Figure A.72: Posts to download when User 4 becomes online (log-log scale)
<table>
<thead>
<tr>
<th>Number of Posts</th>
<th>Total Size (MB)</th>
<th>Interval (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>0.1</td>
<td>100</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td>1000</td>
</tr>
</tbody>
</table>

**Figure A.73:** Posts to download when User 5 becomes online (log-log scale)
Figure A.74: Posts to download when User 6 becomes online (log-log scale)
Figure A.75: Posts to download when User 7 becomes online (log-log scale)
Figure A.76: Posts to download when User 8 becomes online (log-log scale)
Figure A.77: Posts to download when User 9 becomes online (log-log scale)
Figure A.78: Posts to download when User 10 becomes online (log-log scale)
Figure A.79: Posts to download when User 11 becomes online (log-log scale)
Figure A.80: Posts to download when User 12 becomes online (log-log scale)
Figure A.81: Posts to download when User 13 becomes online (log-log scale)
<table>
<thead>
<tr>
<th>Number of Posts</th>
<th>Total Size (MB)</th>
<th>Interval (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.001</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>0.1</td>
<td>100</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
<td>1000</td>
</tr>
</tbody>
</table>

**Figure A.82**: Posts to download when User 14 becomes online (log-log scale)
Figure A.83: Posts to download when User 15 becomes online (log-log scale)
Figure A.84: Posts to download when User 16 becomes online (log-log scale)
Bibliography


