Scalability Issues in Online Social Networks

TAHIR MAQSOOD and OSMAN KHALID, COMSATS Institute of Information Technology
RIZWANA IRFAN, North Dakota State University
SAJJAD A. MADANI, COMSATS Institute of Information Technology
SAMEE U. KHAN, North Dakota State University

The last decade witnessed a tremendous increase in popularity and usage of social network services, such as Facebook, Twitter, and YouTube. Moreover, advances in Web technologies coupled with social networks has enabled users to not only access, but also generate, content in many forms. The overwhelming amount of produced content and resulting network traffic gives rise to precarious scalability issues for social networks, such as handling a large number of users, infrastructure management, internal network traffic, content dissemination, and data storage. There are few surveys conducted to explore the different dimensions of social networks, such as security, privacy, and data acquisition. Most of the surveys focus on privacy or security-related issues and do not specifically address scalability challenges faced by social networks. In this survey, we provide a comprehensive study of social networks along with their significant characteristics and categorize social network architectures into three broad categories: (a) centralized, (b) decentralized, and (c) hybrid. We also highlight various scalability issues faced by social network architectures. Finally, a qualitative comparison of presented architectures is provided, which is based on various scalability metrics, such as availability, latency, interserver communication, cost of resources, and energy consumption, just to name a few.

CCS Concepts: • **Networks → Network types**; *Overlay and other logical network structures*; Online social networks

Additional Key Words and Phrases: Scalability, social network, centralized social networks, decentralized social networks, hybrid social networks

ACM Reference Format:
Tahir Maqsood, Osman Khalid, Rizwana Irfan, Sajjad A. Madani, and Samee U. Khan. 2016. Scalability issues in online social networks. ACM Comput. Surv. 49, 2, Article 40 (September 2016), 42 pages.
DOI: http://dx.doi.org/10.1145/2968216

1. INTRODUCTION

A social network is a platform that facilitates individuals to communicate with each other, connected through social relations, such as family, friends, and colleagues [Nepali and Wang 2014; Symeonidis et al. 2014]. The introduction of Web 2.0, coupled with the social networks, has changed the that way people interact socially. Users not only act as consumers of online data, but are actively involved in the creation of data in the form of videos, photos, and blogs [Chowdhury et al. 2014; Ahmed et al. 2013]. In the last decade, social network services, such as Facebook, Twitter, YouTube, Myspace, and LinkedIn, have witnessed tremendous growth [Kim et al. 2010; Al Mutawa et al. 2012]. For instance, in December 2015, the number of Facebook users exceeded one and

Authors' addresses: T. Maqsood, O. Khalid, and S. A. Madani, Computer Science Department, COMSATS Institute of Information Technology, Abbottabad, Pakistan; emails: {tmaqsood, Osman, madani}@ciit.net.pk; R. Irfan and S. U. Khan, North Dakota State University, Fargo, North Dakota, USA; emails: {rizwana.irfan, samee.khan}@ndsu.edu.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2016 ACM 0360-0300/2016/09-ART40 $15.00
DOI: http://dx.doi.org/10.1145/2968216

a half billion[1]. However, the enormous and rapid growth of social networks gives rise to novel challenges, particularly related to scalability. Scalability refers to the ability of a system to maintain the performance under an increased load. For instance, notifying about status updates and distribution of User Generated Content (UGC) to millions of users in Facebook, as well as dissemination of tweets to a large number of Twitter subscribers in an efficient and timely manner, is possible only if such systems are scalable. Because of the growing number of users and services, social networks need to remain highly scalable to meet future performance requirements [Ugander et al. 2011].

For the past few years, a substantial amount of research has been conducted in the field of social networks [Al Mutawa et al. 2012; Kayastha et al. 2011; Kapanipathi et al. 2011; Nguyen et al. 2011; Kim et al. 2010; Bonifati et al. 2010; Yang et al. 2010]. Numerous surveys have been conducted to explore the different dimensions of social networks, such as security, privacy, data acquisition, anonymous access to data, and encryption techniques for social networks [Bodriagov and Buchegger 2011; Canali et al. 2011; Beach et al. 2009; Zhou et al. 2008; Ellison et al. 2007]. Ellison et al. [2007] have presented the historical perspective and development of social networks and highlighted issues related to information privacy in social networks. Zhou et al. [2008] have presented a survey of privacy preserving anonymization techniques for publishing data in social networks. Beach et al. [2009] have also explored the privacy and security threats faced by social networks, but restricted the analysis to mobile social networks. Canali et al. [2011] surveyed data acquisition techniques for social networks using crawlers and discussed various related privacy issues. Kayastha et al. [2011] have presented a detailed survey of applications, architectures, and different design issues for the mobile social networks. Bodriagov and Buchegger [2011] have highlighted the encryption requirements for the social networks and presented a comparison of various encryption schemes, but have restricted the analysis to peer-to-peer social networks. Recently, Paul et al. [2014] have conducted a survey of decentralized social networks, mainly focusing on different resource provisioning techniques, storage and access control of user data, and user interaction approaches proposed for decentralized online social networks.

Most of these surveys either focused on privacy and security-related issues or other domains, such as anonymization, data acquisition and dissemination, and resource provisioning techniques for the social networks. However, none have explicitly addressed the scalability issues faced by social networks. In this survey, we have presented various significant features of social networks. Moreover, social networks have been categorized into three broad architectures: (a) centralized, (b) decentralized, and (c) hybrid. Various scalability challenges, such as: (a) handling extremely large number of users, (b) infrastructure management, (c) content dissemination, (d) data storage, and (e) energy consumption, faced with these categories, have been discussed in detail. After an extensive literature survey, certain critical parameters have been outlined as scalability metrics, to evaluate the scalability of social networks. Furthermore, a tabular comparison of these categories of social networks has been presented on the basis of identified scalability metrics.

The rest of the article is organized as follows. Section 2 presents an overview and taxonomy of the social networks. Section 3 provides a detailed discussion on scalability issues and challenges faced by the centralized, decentralized, and hybrid social networks. Future directions and challenges faced by the social networks are provided in Section 4. We present our conclusions in Section 5.

[1]http://newsroom.fb.com/company-info/.

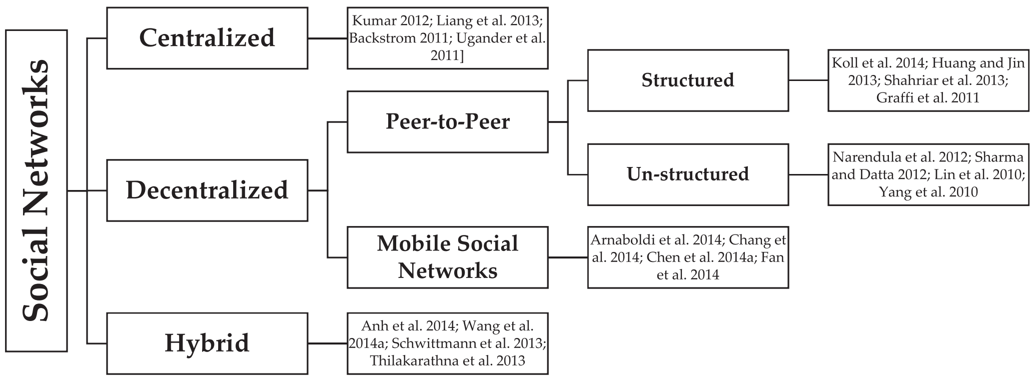

Fig. 1. Taxonomy of social network architectures.

2. SOCIAL NETWORKS

Social networks provide a platform that allows individuals to communicate with family, friends, and colleagues who share social relationships, common interests, and hobbies [Symeonidis et al. 2014; Kim et al. 2010]. Social networks allow users to connect with friends, discover new friends, and share UGCs, such as videos, documents, blogs, and photos [Del Giudice et al. 2014; Kim et al. 2010]. According to Basuchowdhuri et al. [2014], a social network can also be represented by a set of connected graphs consisting of nodes (individuals) that are connected to each other through edges (relationships).

Social networks have some significant characteristics. One of the characteristics is the support for multiple communication modes, including one-to-one, one-to-many, and many-to-many. For instance, online chat (e.g., Facebook chat NaP10) is an example of one-to-one communication. An example of one-to-many and many-to-many communication is Twitter, in which a user can send tweets to many followers. Similarly, Twitter allows multiple individuals to engage in a conversation about a specific topic. The second distinct characteristic of social networks is the dynamic expansion of users. For instance, according to Ugander et al. [2011], each Facebook user has an average number of friends of around 200, and this number is expected to increase significantly in the future.

Based on the factors, such as communication model, users' control on the profile data and UGCs, and presence or absence of a centralized authority, social networks are categorized into three broad architectures: (a) centralized, (b) decentralized, and (c) hybrid. A detailed taxonomy of social network architectures is presented in Figure 1.

2.1. Centralized Social Networks

In centralized social networks, users typically connect to and use various services of social networks through a Web browser, such as Internet Explorer, Firefox, and Chrome. The centralized social networks constitute a Wide Area Network (WAN), and are supported by a large number of servers located in a single building or distributed across cities or countries. The centralized social network architecture refers to the online social networking services, such as Facebook, Twitter, Google+, YouTube, Instagram, and MySpace. In addition to Web-based access, the majority of the centralized social network service providers offer mobile smartphone applications. Mobile applications allow users to connect with social networking services using mobile phones anywhere anytime through the Internet. However, the aforementioned social networking services are categorized as centralized, because these are based on client–server architecture and are controlled by a single administrative authority [Chowdhury et al. 2014;

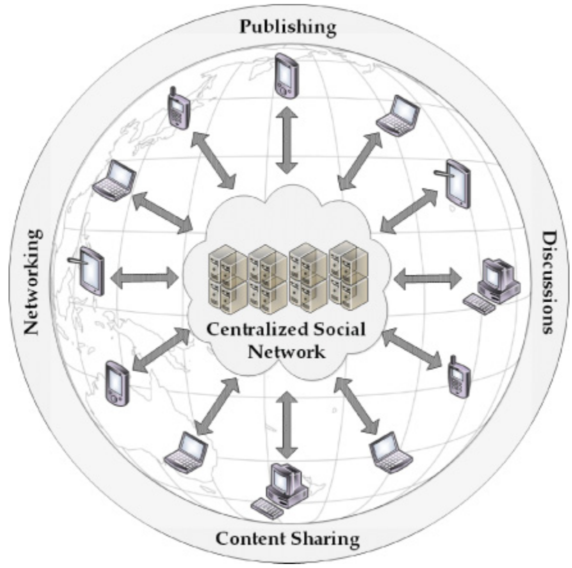

Fig. 2. Centralized social network.

Kryczka et al. 2010; Stuedi et al. 2011]. Moreover, all the users' profile information and UGCs are stored at servers, owned and operated by a single service provider that has complete control over all users' data [Kryczka et al. 2010; Stuedi et al. 2011]. Figure 2 shows the overview of a centralized social network in which users connect and share UGC through a centralized social network service provider over the Internet.

2.2. Decentralized Social Networks

Decentralized social networks combine social and decentralized elements. Such social networks do not depend on any centralized server and are not controlled by any administrative authority. In contrast to online social networks, in which the service provider has to bear all the costs in providing services, the decentralized online social network uses the storage space of its participants to increase the availability or the survivability of the data. The components of a decentralized social network generally consist of a network of trusted servers or peer-to-peer systems on a distributed information management platform. A user is allowed to communicate and share contents directly with other users without joining any centralized infrastructure service such as Facebook or MySpace [Chowdhury et al. 2014]. Decentralized social networks have been further split into two categories: (a) Peer-to-Peer (P2P) socials networks [Nguyen et al. 2011; Yang et al. 2010; Bonifati et al. 2010] and (b) mobile social networks [Koll et al. 2014; Huang and Jin 2013; Sarigol et al. 2010]. A detailed description of these categories is presented in the subsequent text.

2.2.1. Peer-to-Peer Social Networks. P2P social networks allow users to communicate and share contents directly without the intermediation and control of central server and mandatory requirement of Internet connectivity [Kapanipathi et al. 2011; Bonifati et al. 2010]. Moreover, P2P networks enable the participants to share resources, such as computational power, storage, and bandwidth [Androutsellis-Theotokis and Spinellis 2004]. The P2P social network is based on a P2P network rather than relying on a central infrastructure or authority [Raji et al. 2014]. P2P systems have the ability to self-configure, especially in the case of transient failures. In such cases, the P2P systems are able to maintain an acceptable level of performance and ensure

Table I. Comparison of P2P Social Networks

Scheme	Application Domain	Open Source	Privacy	Control over UGC	Require Internet	Support new Applications	Encryption	Censorship	User Authentication
[Koll et al. 2014]	Social network	✓	✓	✓	✗	✓	✓	✗	✓
[Huang and Jin 2013]	Microblogging	✗	✗	✗	✗	✗	✗	✗	✗
[Narendula et al. 2012]	Social network	✗	✓	✓	✗	✗	✗	✗	✗
[Sharma and Datta 2012]	Social network	✗	✓	✓	✗	✗	✓	✗	✗
[Kapanipathi et al. 2011]	Microblogging	✓	✓	✓	✗	✗	✗	✗	✗
[Nguyen et al. 2011]	Live streaming	✓	✗	✗	✓	✗	✗	✗	✗
[Graffi et al. 2011]	Social network	✗	✓	✓	✓	✓	✓	✗	✓
[Bonifati et al. 2010]	Social network	✗	✓	✓	✗	✓	✓	✗	✓
[Yang et al. 2010]	Social interaction	✗	✗	✗	✗	✗	✗	✗	✗
[Kourtellis et al. 2010]	Social network	✗	✓	✓	✗	✓	✓	✗	✓

connectivity [Rodrigues and Druschel 2010]. Some major advantages of using P2P technology are (a) a high degree of decentralization, (b) self-organization, (c) fault tolerance, and (d) abundance and diversity of resources [Rodrigues and Druschel 2010].

During the last few decades, social networks and P2P technology have evolved in parallel, both experiencing tremendous growth. P2P technology has evolved as a promising platform for the sharing of large-scale data and multimedia content [Rodrigues and Druschel 2010]. Recently, P2P networks have been used extensively for file sharing, such as in Gnutella, Napster, and BitTorrent. Successful implementation of P2P networks has also been observed in various other disciplines, such as large-scale multimedia streaming, video on demand, and social networks [Rodrigues and Druschel 2010]. Table I contains a comparison of some of the selected P2P social networks proposed in the literature.

One of the reasons that limits the wide adoption of P2P social networks is the lack of proper incentives for the users to share their resources. Therefore, an effective mechanism needs to be devised to offer proper incentives to the users of P2P social networks. In this regard, Sharma and Datta have presented a super-peer based hybrid social network termed as "SuperNova" [Sharma and Datta 2012]. SuperNova takes advantage of the fact that different peers, participating in the P2P network, have different capabilities in terms of computation and storage capacity. Therefore, the highly capable peers can take on the role of super-peers. The super-peers help to find new friends, provide additional storage, and give reference to the nodes that maintain directories of users and content to efficiently locate users [Sharma and Datta 2012]. The incentives to become a super-peer are multifold, such as (a) promoting P2P social networks, (b) becoming an influential member, and (c) monetizing in order to get monetary benefits, such as running advertisements. Moreover, users that are not super-peers can have different motivations, such as privacy preservation, complete control over their own data, and to become an active member of a social network.

2.2.2. Mobile Social Networks. In the last few years, mobile technology has seen unprecedented growth and the use of mobile phones has increased manyfold [da Costa et al. 2014; Lubke et al. 2011]. In mobile social networks, users are able to create social networks "*on-the-fly*" at public locations, such as airports, restaurants, stadiums, and

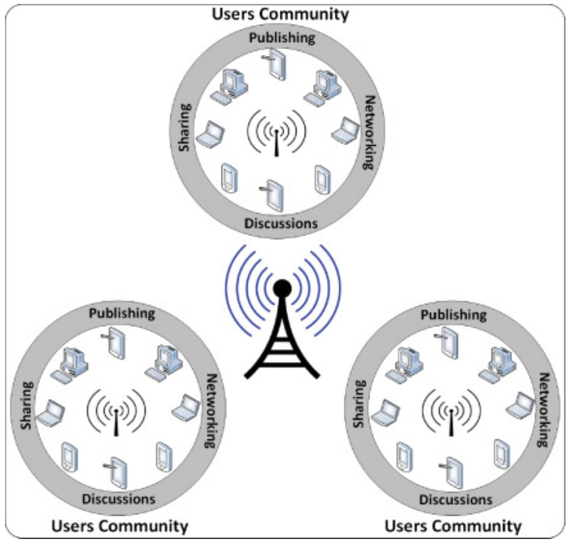

Fig. 3. Different communities of users establishing mobile ad hoc social network.

Table II. Comparison of Mobile Social Networks

Scheme	Application Domain	Open Source	Privacy	Control over UGC	Require Internet	Support new Applications	Encryption	Censorship	User Authentication
[Chang et al. 2014]	Social network	✗	✓	✓	✓	✗	✗	✗	✗
[Arnaboldi et al. 2014]	Social network	✗	✗	✗	✗	✓	✗	✗	✗
[Li et al. 2012a]	Social network	✗	✓	✓	✗	✗	✓	✗	✗
[Sarigol et al. 2010]	Social interaction	✗	✗	✗	✗	✓	✗	✗	✗

during events, for instance, conferences, ceremonies, exhibitions, or office meetings [Chang et al. 2014; Bellavista et al. 2013; Li et al. 2012a]. In mobile social networks, users connect to a social network through mobile phones. Mobile social networks allow users to communicate and share content with other mobile users in the physical proximity using Bluetooth or Wi-Fi technology without using the Internet or cellular infrastructure [Chang et al. 2014; Bellavista et al. 2013], as presented in Figure 3.

New-generation mobile smartphones are equipped with more processing power, memory, and storage capacity that lead to the development of many new and interesting social networking applications, such as FourSquare[2], WhatsApp, and location-based gaming. Table II depicts the comparison of some of the mobile social networks proposed in the scientific literature.

[2]https://foursquare.com/.

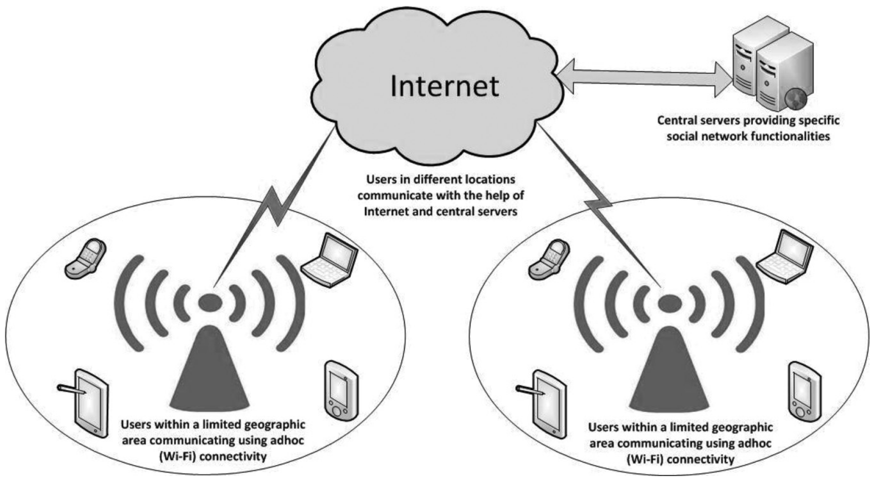

Fig. 4. Hybrid social network.

2.3. Hybrid Social Networks

Hybrid social networks combine the features of centralized and decentralized social networks. In the hybrid social networks, users can communicate and share contents with other users either using centralized servers and/or using decentralized systems, such as P2P. The decentralized part of hybrid social networks provides various services, such as content distribution, file sharing, and location-based searching in a decentralized manner. Alternatively, the centralized infrastructure is utilized in support of a decentralized setup to provide services, such as user authentication, content indexing, and storing user profiles. Figure 4 presents an example of a hybrid social network in which users can either communicate directly using Wi-Fi connectivity or through centralized servers over the Internet.

Similarly, some hybrid social networks are supported by Content Delivery Networks (CDNs) or can be peer-assisted to enhance the scalability and availability of social networks [Chard et al. 2012]. For instance, Kryczka et al. [2010] have presented a hybrid social network that stores UGCs at user devices and the social network service provider only hosts the required applications and services, such as user authentication and content indexing. Similarly, [Anh et al. 2014] have proposed a middleware, called Mosco, for developing mobile social applications that uses cloud-based storage for scalable data management.

3. SCALABILITY IN SOCIAL NETWORKS

According to Chakradhar and Raghunathan [2010], scalable computing systems maintain consistent performance under an expanded workload. A computing system refers to the set of computational resources that include processing power, storage capacity, and networking. For instance, a network is scalable if expanding the number of nodes in the network does not degrade the overall performance and the average throughput of the system remains consistent [Alajlan and Elleithy 2014]. In the case of social networks, the scalable social network should not compromise on performance and maintain the same level of latency and response time even if more contents are added and more users connect to the social network simultaneously.

Social networks have become an integral part of the Internet community. Due to the rapid growth of social network users, maintaining high performance in terms of user experience and service response time is a crucial task [Pujol et al. 2011]. In the case of social networks, not inculcating scalability in the initial design can be disastrous due to the exponential growth of social network users. For instance, during the period of 1 year (February 2008 to February 2009), Twitter witnessed 1,382% growth in addition to their usual growth rate [Sakaki et al. 2010]. The successive growth of Twitter resulted in significant downtime due to the fact that the existing architecture of Twitter could not handle the huge amount of network traffic generated by the increased number of users [Pujol et al. 2011]. Consequently, the entire architecture of Twitter was redesigned [Pujol et al. 2011]. Similarly, MySpace[3] developers repeatedly redesigned their Web-based applications, database management systems, and storage systems to handle the rapid growth of users. Based on these facts, it can be safely stated that scalability is very crucial for social networks to provide consistent performance, especially in the case of rapid increase in the number of users and UGCs.

3.1. Distinguishing Features of Online Social Networks

In this section, we briefly mention some distinctive features of online social networks that set them apart from other Internet applications, such as surfing, e-mail, searching, music, videos, and gaming.

3.1.1. Dynamic Social Graph. One of most distinguishing features of social networking applications is the dynamic social graph. From a social network user's perspective, a social graph consists of a set of nodes in which each user represents a node and an edge between any two given nodes indicates a friendship relation [Jin et al. 2013]. As new users keep joining the social network and new friend relationships are created, the social graph is updated. Frequent creation of social links among users and the resulting dynamics of the social graph dramatically changes the infrastructure requirements of a social networking service in terms of computation, storage, and retrieval of content [Yuan et al. 2012]. For instance, depending on the impact of a change in a social graph, configuration of underlying servers can be updated in terms of how data can be partitioned, or how much and when to replicate data to maintain low response time [Gaito et al. 2012]. Due to the dynamic nature of a social graph, it is very challenging to predict the expansion of a social graph considering the large number of users and rapid growth of data [Jin et al. 2013]. In contrast, traditional cloud-based web applications such as eBay, Yahoo, and Amazon are easier to scale for two reasons. First, on average, only a small percentage of users are active on a site at one time. Consequently, such sites comparatively require less RAM cache and processing power to handle all the active users. Second, the users mostly access their own and commonly cached data. Alternatively, on Facebook, if a person with 200 plus friends signs in, Facebook has to gather the status of all 200 friends, which means 200 requests need to go out simultaneously and the replies need to be merged. As friends' data is active all the time, everything must be kept in RAM cache, which makes it hard to partition a social networking system.

3.1.2. Frequency of Content Generation. In traditional websites or other Internet applications, the amount and type of content is controlled by the service provider, whereas in social networks, the content is generated and consumed by users. For instance, according to Li et al. [2012b], over 500 links of YouTube videos are shared by Twitter users per minute and more than 150 years duration of YouTube videos are watched daily on Facebook. Moreover, a majority of actions performed by social network users constitute actions, such as likes, tags, or comments that are frequently accessed and updated on social network servers. These user actions generate a major proportion of

[3]http://www.baselinemag.com/c/a/Projects-Networks-and-Storage/Inside-MySpacecom/.

network traffic due to the large number of small data packets transmitted over the Internet [Mathieu et al. 2012; Pallis et al. 2011].

3.1.3. Unpredictable Expansion. Millions of users access various social network services, whose usage and interaction patterns are extremely difficult to predict. Services provided by social networks are dynamic in nature as new plug-ins and features are constantly introduced into the system. In such cases, sudden load spikes can occur, for instance, when a new plug-in is deployed or a new feature becomes popular with users. Moreover, users of social networks are distributed throughout the world, organized in tightly coupled communities of highly interactive people. Consequently, load spikes can be generated at different times from different parts of the world that make scalability of social network services harder [Buyya et al. 2010].

3.1.4. Always-On Connectivity. With the proliferation of mobile smartphones, almost all social networks provide mobile applications to seamlessly connect with and use social network services [Cui and Honkala 2011]. However, the users that stay continuously connected through mobile applications put additional load on social network servers and data carriers because updates and notifications are instantaneously pushed to the users. This is certainly in contrast with other traditional Internet applications, such as surfing, searching, online shopping, listening to music, and watching movies. In the applications mentioned earlier, users connect and utilize the services of a particular Internet application of a website when required; afterward, the session is closed to free server resources.

3.1.5. Asymmetric Relationships. In large centralized social networks, such as Facebook and Twitter, some users have an extremely large number of friends or followers, while the average users may have significantly fewer relationships. For instance, in Twitter, approximately 50 users have more than one million followers, whereas the average number of followers for Twitter users is 208 [Lamba et al. 2015]. Due to the directed nature of the Twitter "*follow*" relationship, Twitter users have low reciprocity, that is, 22% of users have reciprocal relationships [Kwak et al. 2010].

Based on distinguishing features of online social networks discussed earlier, we have identified some critical scalability challenges faced by each category of social network—centralized, decentralized, and hybrid—that are discussed in subsequent sections, respectively.

3.2. Scalability Issues in Centralized Social Networks

3.2.1. Large Number of Highly Connected Users. Rapid growth in number of users that ubiquitously use social networking services creates a serious scalability challenge for large centralized social network service providers, such as Facebook [Nishtala et al. 2013; Menon 2012; Thusoo et al. 2010]. For instance, according to Thusoo et al. [2010], the amount of compressed data that is loaded daily by Facebook increased to 10TB to 15TB from 5TB to 6TB in only six months due to the rapid increase in the number of Facebook users in 2010. The amount of data ingested daily into Facebook reached over 500TB in 2012 [Menon 2012]. To make matters worse, certain users, such as celebrities, have an extremely large number of fans or followers compared to general users, intricating scalability. For instance, the tweet and retweet from a celebrity with millions of followers generates more traffic as compared to a tweet from a common user having hundreds of followers. According to Kwak et al. [2010], any retweet reaches on average 1,000 users regardless of the number of followers of person that originated the tweet.

According to Xu et al. [2011], approximately 50% of Twitter users have 10 or less followers, 90% of users have 100 or fewer followers, and 95% of users are followed by

Table III. User Statistics of Social Networks

	Facebook	Google+	Twitter	LinkedIn	Instagram	Pinterest	YouTube
Number of monthly active users (millions)	1,590[4]	300[5]	320[6]	400[7]	400[8]	100[9]	1,000+[10]

less than 200 followers. Moreover, only 0.1% of users have more than 10,000 followers. According to Backstrom [2011], the percentage of Facebook users with less than 10 friends is 10%, while 20% of users have less than 25 friends and, on average, 50% of users have more than 100 friends . Ugander et al. [2011] report that approximately 70% of users have less than 200 friends. Moreover, approximately 10% of users have more than 1000 friends. Similarly, degree distribution of Facebook users indicates that the degree distribution of global and U.S. Facebook users is quite similar [Ugander et al. 2011].

Table III presents the statistics related to number of users in some of the large social networks. However, the large number of users is not the only challenge that limits scalability of centralized social networks. The real-time social graph, which represents relationships among social network users, is another reason behind the difficulty in scaling centralized social networks. Scaling centralized social networks is more difficult than traditional networks and other Web-based applications due to the tight coupling between different communities of social network users. For instance, when a user visits traditional websites, such as MSN and Yahoo, the Web server fetches the profile of the user with a single query and generates a customized view according to the preferences of users [Farrington and Andreyev 2013]. However, centralized social networks, such as Facebook and Twitter, have millions of users with hundreds of friends. The social network server needs to fetch the profile of users along with profile and status updates of all the friends in the users' contact list in order to generate a customized and updated view [Farrington and Andreyev 2013]. Facebook and Twitter distribute users' data across hundreds and thousands of servers. Consequently, to get the status updates of users' friends, all servers containing users data need to be queried [Yuan et al. 2012]. To serve a page to a user, hundreds of machines need to examine thousands of data elements [Kumar 2012]. Therefore, the response times can be variable based on multiple factors, such as network condition and server load. In such cases, the maximum delay observed by the users will be the delay incurred at the server having highest latency [Pujol et al. 2011].

In social networks, neighboring nodes of users are constantly queried to generate updated content [Kumar 2012]. Therefore, even when only a small fraction of social network users are online at a particular instance, the majority of the users' data is being accessed all the time and is kept active [Kumar 2012]. Moreover, it is hard to partition and denormalize social network data due to the highly interconnected nature of the social graph [Joshi 2013]. According to Nishtala et al. [2013], at Facebook, a single read operation requires fetching data from a variety of systems, such as Hadoop Distributed File System (HDFS) installations, MySQL databases, and backend services. For instance, the *memcached* system, an open-source customized caching solution, receives more than a billion requests every second at Facebook and stores

[4]http://newsroom.fb.com/company-info/ (Accessed on: March 1, 2016).
[5]http://expandedramblings.com/index.php/google-plus-statistics/ (Accessed on: March 1, 2016).
[6]https://about.twitter.com/company (Accessed on: March 1, 2016).
[7]https://press.linkedin.com/about-linkedin (Accessed on: March 1, 2016).
[8]http://instagram.com/press/ (Accessed on: March 1, 2016).
[9]http://fortune.com/2015/09/17/pinterest-hits-100-million-users/ (Accessed on: March 1, 2016).
[10]https://www.youtube.com/yt/press/statistics.html (Accessed on: March 1, 2016).

trillions of items. A single-user request for a popular Facebook page can result in hundreds of get requests, on average 521 requests, to *memcached* servers, where each request may be served by a different *memcached* server in the cluster [Nishtala et al. 2013]. Consequently, a tremendous amount of queries and network traffic is generated when millions of users access various types of social network services, such as tweets, status updates, photos, and videos upload. In contrast to traditional websites, almost all data of a social network must be kept active all the time. In centralized social networks, handling a huge amount of status, profile updates, and UGCs, as well as ensuring availability and consistency of the uploaded data, creates a significant scalability challenge.

3.2.2. Infrastructure Issues. Large centralized social networks, such as Facebook and Twitter, require huge infrastructure for thousands of servers, housed in large data centers, needed to (a) support various services, such as dissemination of real-time status updates and UGCs; and (b) meet performance requirements. A data center is considered to be a collection of a large number of computing resources, usually made up of commodity hardware connected through high-speed network [Greenberg et al. 2008; Bilal et al. 2014; Hammadi and Mhamdi 2014]. Although the infrastructure issues are common as well in other cloud-computing applications, for example, web mail services, online shopping websites, or Google Maps, resource demand in social networks is much higher due to frequent dynamic social-graph creation and a larger number of queries floating over the social graph.

A data center is simply a collection of interconnected computing resources. Efficient utilization of data center resources largely depends on the type of application(s) being hosted. A significant proportion of applications hosted in data centers belong to the cloud-computing domain. The major difference between a data center that hosts common cloud-computing services and the data centers used to provide social networking services is in the intensity of resource utilization and user access pattern. According to Ghribi [2014], average resource utilization in most data centers is as low as 12%. Similarly, average server utilization in data centers remained between 12% and 18% for the duration 2006 to 2012 [Ghribi 2014]. On the other hand, the majority of the resources, including servers, in a social network data center remain constantly active to provide uninterrupted services to social network users. For instance, when Facebook application servers receive a user request for a page, on average, 130 additional requests are generated to fetch data from internal servers to construct the required HTML page [Ousterhout et al. 2011]. The requirements to fetch each data object, multiplied by millions of users active at any given time, lead to significant challenges in designing the underlying databases, caches, and network [Bronson et al. 2015]. Due to an extremely large number of social network users spread throughout the world, access patterns, usage timing, and resource utilization of centralized social networks is significantly higher than common cloud data centers. According to Chowdhury et al. [2014], approximately 80% of active Internet users visit at least one of the social networking sites daily. Similarly, Global Web Index[11] reported that Internet users spend, on average, 109min per day on social networks or messaging services. Consequently, in contrast to traditional cloud-computing applications, a significant proportion of social network users' data needs to be kept active in caches at all times, as currently being practiced by Facebook using *memcached* and TAO [Bronson et al. 2013; Nishtala et al. 2013].

In a recent study conducted at Facebook, it was reported that traffic and workload characteristics observed at Facebook significantly differ from existing data center traffic studies reported in the literature [Roy et al. 2015]. This is mainly due to various

[11]http://www.globalwebindex.net/blog/social networks-grab-a-third-of-time-spent-online.

essential services required to be supported by social networking infrastructure that include (a) providing near real-time interaction among users, (b) aggregating a variety of content from multiple sources instantaneously, (c) enabling fast access and updates of shared popular content, and (d) scaling to handle hundreds of millions of requests per second. Consequently, large social networks present significant challenges in terms of increased computational, I/O processing, and network demands to the underlying infrastructure [Nishtala et al. 2013]. In this section, we discuss the data center infrastructure-related issues, specifically considering operational characteristics of centralized social networks.

To meet the continuously increasing resource requirements, Facebook deployed five data centers. Four of the data centers are located in different cities of the United States and one is located in Sweden [Miller 2013]. Similarly, Twitter has two data centers located at different regions of the United States. Operating such a huge infrastructure is very expensive for social network service providers. Here, we highlight various components that contribute to infrastructure-related expenses required to operate large data centers.

(1) *Cost of Equipment:* Due to the rapid expansion of social network users, the underlying infrastructure of social networks also need to be constantly upgraded. Therefore, the number of servers that provide social network services range from a few servers to thousands of servers. The number of Facebook server nodes increased 260 times from 2008 to 2012 [Menon 2012]. For instance, Facebook had more than 180,000[12] and LinkedIn had 30,000 servers[13] till September 2013. Moreover, the cost of network equipment and cabling also needs to be considered. According to Kliazovich et al. [2013] and Greenberg et al. [2008], 45% of the data center cost is attributed to servers, 25% to infrastructure, 15% to network equipment, and 15% to power systems.

(2) *Operational Expenses:* The infrastructure for large centralized online social networks is composed of thousands of servers located in multiple data centers that may be distributed geographically [Kryczka et al. 2010]. The large number of servers is connected through network links, routers, and switches. Therefore, server components and network equipment fail at regular intervals [DeCandia et al. 2007]. The large infrastructures require continuous troubleshooting and maintenance to diagnose and repair faults. Well-trained and skilled staff is required to manage such a huge infrastructure. Koomey et al. [2007] report that 1/4 of the annual costs of data centers are attributed to the operational expenses and 3/4 is attributed to capital expenditures that include infrastructure and server costs. According to Ebrahimi et al. [2014], the construction cost and annual operational expenses of a typical data center are approximately \$15000/meter^2 and \$1500/meter^2, respectively. Another major source of operational expenses for large online social network service providers, such as Facebook and Google, is bandwidth cost. For instance, the daily expense of YouTube for server bandwidth is one million dollars [Cheng et al. 2013].

(3) *Energy Consumption:* The centralized social network infrastructure holds a large number of servers that consume a significant amount of energy [Kryczka et al. 2010]. In addition to the electricity that is used to power the servers, an approximately equal amount of energy is consumed by the Heating, Ventilation, and Air Conditioning (HVAC) system for cooling the servers and networking equipment [Kryczka et al. 2010]. Moreover, within Google data centers, network equipment

[12]http://www.datacenterknowledge.com/archives/2012/08/15/estimate-facebook-running-180000-servers/.
[13]http://www.enterprisetech.com/2013/11/07/linkedin-copes-server-explosion-revved-cfengine/.

consumes approximately 20% of total power in the situation in which servers are being utilized at 100%. However, the proportion of power consumption for network equipment rises to 50% when servers are utilized at 15% [Wang et al. 2014c]. Therefore, the cost of energy used to power the centralized social network infrastructure is an important factor. According to Wang et al. [2014b], the cost of energy consumption can be up to 50% of the total operational expenses of the data center. For instance, according to estimates taken in October 2008, Facebook spent approximately two million dollars per month to power the servers and for cooling purposes [Kryczka et al. 2010]. Similarly, Facebook data centers consumed 678 million Kwh of energy in 2012, a 33% increase compared with the energy consumption in 2011 [Miller 2013]. Moreover, the carbon footprint of Facebook, as a result of CO_2 emissions, increased 52% in 2012 compared to 2011 [Miller 2013]. Similarly, Google and Microsoft spend $67 million and $36 million in terms of annual electricity bills for their data centers [Deng et al. 2014].

This discussion shows that a significant amount of energy and cost is incurred for the development and maintenance of the large infrastructure needed to operate large centralized social networks. Moreover, the energy consumption leads to a substantial increase in greenhouse gas emissions that are considered “not environment friendly” [Wang and Khan 2013]. The costs related to energy consumption and maintenance will increase as centralized social networks grow. Keeping in mind this discussion, the continuously expanding infrastructures, maintenance, and operational costs with high energy requirements can affect the scalability of centralized social networks.

Scientific literature suggests two solutions for scaling the infrastructure of large-scale computing systems, such as social networks. The two solutions are (a) vertical scaling and (b) horizontal scaling. Vertical scaling, also referred to as scaling up, allows addition of computational resources, such as processors, memory, and storage, to existing servers or virtual machines or acquiring new servers with the enhanced capacity [Yang et al. 2014; Garg et al. 2012]. However, acquiring a completely new server can be a very expensive solution due to the high cost of specialized servers [Garg et al. 2012]. Horizontal scaling, also referred to as scaling out, is achieved by adding more servers, usually low-cost commodity machines, to the existing pool of available resources [Pujol et al. 2011]. In the case of cloud computing, horizontal scaling refers to the process of adding more virtual machines to the existing pool of resources [Yang et al. 2014]. However, horizontal scaling requires partitioning of the data across multiple servers, which leads to increased communication among servers due to the tight coupling of each social network user with multiple interconnected communities [Pujol et al. 2011].

To meet continuously increasing service demands, large centralized social network service providers, such as Facebook, Twitter, and LinkedIn, have adopted horizontal scaling [Thusoo et al. 2010]. As a result, social network service providers experienced a continuous increase in the number of servers in their data centers. For instance, the number of Facebook servers increased to 180,000[14] in 2012 compared to 30,000[15] servers in the year 2009. For centralized social networks, horizontal scaling has several benefits, such as (a) the convenience of adding new commodity servers compared to upgrading high-end servers, (b) the cost of purchasing commodity servers is much less than upgrading a high-end server, (c) maintenance or upgrading can be carried out without disrupting the service, and (d) the load can be distributed among a large number of servers that can be located in geographically distributed data centers.

[14]http://www.datacenterknowledge.com/archives/2012/08/15/estimate-facebook-running-180000-servers/.
[15]http://www.datacenterknowledge.com/archives/2009/10/13/facebook-now-has-30000-servers/ (Accessed on: March 1, 2016).

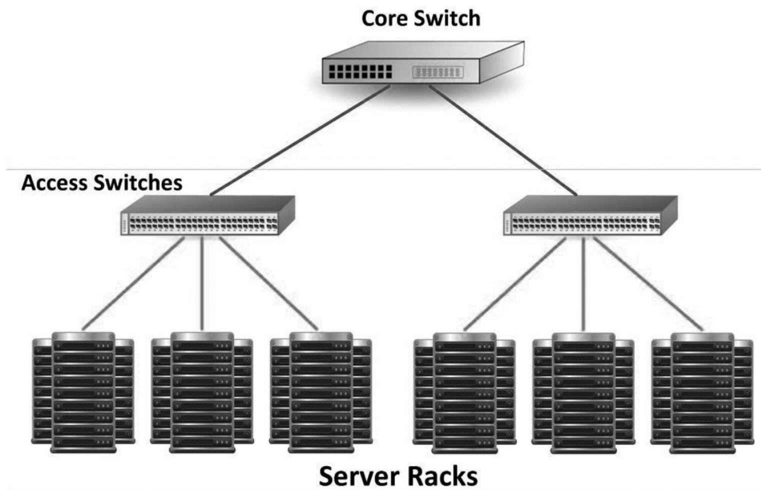

Fig. 5. An example of a data center network.

Readers interested in exploring the trade-off between horizontal and vertical scalability are encouraged to study the work of Garg et al. [2012].

3.2.3. Internal Network Traffic. Network interconnection among the large number of servers and the resulting internal traffic is another scalability challenge faced by the centralized social networks, such as Facebook, Twitter, and YouTube. Internal traffic refers to the communication that takes place among the social network servers and is initiated within the network. The majority of internal traffic is generated by various services, such as (a) the friend recommendation, (b) real-time notifications, and (c) personalized marketing. Some other services that run at the back end of a social network, such as (a) replication, (b) maintenance, and (c) synchronization of indexes, also contribute to internal network traffic [Farrington and Andreyev 2013]. The internal traffic of Web-based social networks is growing exponentially and has far exceeded external traffic [Farrington and Andreyev 2013; Xu et al. 2011]. External traffic includes the traffic to and from social network users over the Internet. According to Farrington and Andreyev [2013], for 1KB of external traffic received in the form of user requests, 930KB of internal data center traffic is generated.

As mentioned in Section 3.2.2, most centralized social network services are hosted at data centers. Data center networks connect the large number of servers using commodity network equipment based on certain topologies, such as three-tier [Greenberg et al. 2008], fat-tree [Al-Fares et al. 2008], BCube [Guo et al. 2009], and VL2 [Greenberg et al. 2011]. A data center can be located at a single large building or can be geographically distributed across cities and countries [Bird et al. 2014; Chen et al. 2014b]. Most data centers are based on three-tier network architecture, which is shown in Figure 5. In three-tier architecture, all internal traffic generated by servers located in different layer-2 domains needs to traverse the core layer [Greenberg et al. 2008]. In such cases, the utilization of core links is comparatively higher. Consequently, the increase in internal network traffic can lead to frequent congestion of a significant number of core links [Hammadi and Mhamdi 2014; Benson et al. 2010; Greenberg et al. 2008]. Network devices, such as routers and switches, become more occupied to handle internal network traffic, which leads to increased latency and poor response time for social network users. Moreover, according to Bilal et al. [2013], 70% of network bandwidth is consumed by intra–data center communication.

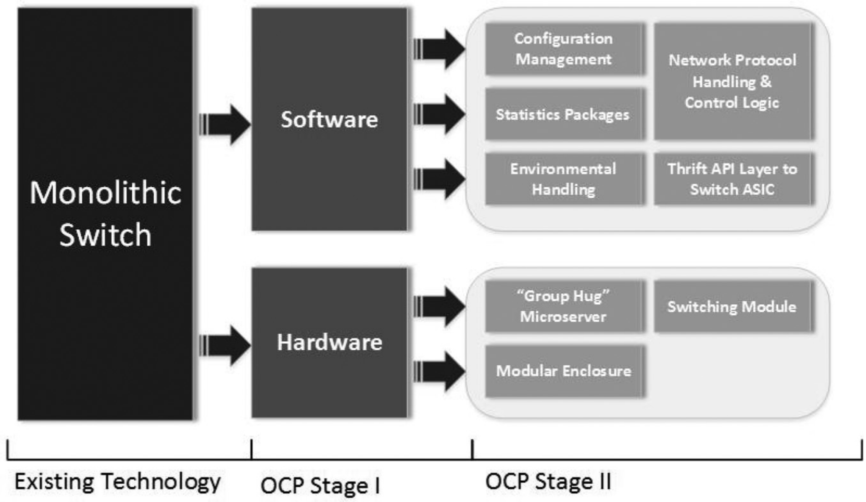

Fig. 6. Overview of modular networking design by Facebook [Bachar 2014].

The underlying network infrastructure of centralized social networks needs to be upgraded to circumvent the increased internal network traffic. A scalable and robust network infrastructure needs to be designed that will not only handle current workloads but can also accommodate the exponential growth of internal network traffic. For instance, during the Football World Cup (2010), Twitter experienced record usage of the service [Xu et al. 2011]. Consequently, to ensure availability and consistent performance, Twitter doubled the capacity of its internal network[16]. Similarly, Facebook had also upgraded its internal data center network architecture and deployed 10 Gbps LAN within the data center [Farrington and Andreyev 2013].

To overcome the deficiencies of existing data center networks, Facebook has completely redesigned its internal data center network. Facebook has adopted the approach of disaggregating traditional data center network components to build a new network system that is more efficient, flexible, and scalable. Facebook has achieved this by designing a top-of-rack (TOR) switch named "Wedge" that runs a Linux-based operating system named "FBOSS" [Bachar 2014]. The objective was to decouple the hardware and software functions of a traditional network stack into basic functions to achieve new levels of automation, visibility, and control. More recently, to overcome the limitations of current monolithic access switches, Facebook designed the first open modular switch, named "6-pack" [Bachar 2015]. The *6-pack* switch can provide the efficiency, flexibility, and scalability required to handle extremely fast and large volumes of data generated by large social network service providers, such as Facebook. All these technologies, developed by Facebook, are part of the open compute project (OCP). These technologies not only benefit the social network service providers but can also be leveraged by other large data center operators as well as network operators at any scale. Figure 6 provides an overview of the development of modular architecture designed by Facebook as part of the OCP.

Similarly, there are numerous proposals from the research community to overcome performance limitations, such as oversubscription and traffic congestion of traditional tree-like DCN architectures. For instance, switch-centric fat-tree [Al-Fares et al. 2008] and VL2 [Greenberg et al. 2011] topologies are proposed to improve the

[16]https://blog.twitter.com/2010/twitter-performance-an-update.

worst-case performance of DCN through overprovisioning of links and switches. However, these topologies result in higher cabling complexity and incur high operational cost [Hamedazimi et al. 2015; Hamza et al. 2015]. Similarly, server-centric topologies, such as DCell [Guo et al. 2008], BCube GuL09, and Ficonn LiG09, are proposed to overcome the limitations of traditional tree-based and switch-centric topologies. However, server-centric topologies put additional workload on servers to carry out networking functions, such as routing and relaying packets. Server-centric topologies may not prove to be the best alternative for centralized social networks in which servers are already engaged in heavy processing. The ever-increasing communication volume and higher-energy consumption of the copper-based Ethernet motivated researchers to explore wireless technologies for DCNs. There are two competing technologies for wireless communication in DCNs: (a) Radio Frequency (RF) and (b) Free Space Optics (FSO). Various proposals, such as Flyways [Halperin et al. 2011], 3D Beamforming [Zhou et al. 2012], and completely wireless Cayley datacentre [Shin et al. 2013] discussed the deployment of 60GHz RF technology in DCNs. However, certain limitations of 60GHz RF wireless technology prohibit its practical implementation in DCNs, including (a) low practical bandwidth, (b) high attenuation and propagation losses, (c) line-of-sight requirement, and (d) interference due to a large number of wireless modules in close proximity [Hamza et al. 2015]. Alternatively, FSO uses optical signals for communication. Initial proposals, such as Helios [Farrington et al. 2011] and c-Through [Wang et al. 2011], proposed hybrid optical/electric switch architectures. Recent proposals, such as FireFly [Hamedazimi et al. 2015], FSO-DC [Hamza et al. 2014], and power-smart indoor optical wireless [Riza and Marraccini 2012], embrace all-optical designs. The advantages of using optical technology are high bandwidth, low power consumption, lower interference compared with RF, and low latency due to high speed of light. For a more detailed discussion on DCN architectures, see Bilal et al. [2014] and Hammadi and Mhamdi [2014].

3.2.4. User-Generated Content Management and Dissemination. Most of the activities performed by social network users are related to creation and sharing of multimedia content, such as videos and photos [Marques and Serrão 2014; Loupasakis et al. 2011]. For instance, YouTube is gaining popularity due to the dissemination of versatile, multicultural, and informative videos. According to Brodersen et al. [2012], more than 3 billion videos are being watched daily on YouTube. Similarly, according to TechCrunch[17], approximately 2.5 billion pieces of information, 2.7 billion likes, 300 million photos, and more than 500TB of data are being added to the Facebook every day. Similarly, the average number of tweets sent daily on Twitter has reached 500 million[18], and 2.1 billion search queries are executed by the Twitter search engine[19]. According to statistics, on Facebook, approximately one million links are shared, two million friend requests are made, and three million messages are sent on Facebook every 20min[20].

Efficient handling and dissemination of the large amount of UGCs creates a significant scalability challenge for large centralized social networks. For instance, Twitter service suffers from poor availability in terms of service rejection rate. In the normal usage scenario, the average rejection rate of Twitter was around 10%, which increased to 20% during the extremely high load during the FIFA World Cup 2010 [Xu et al. 2011]. Moreover, high usage during the World Cup also affected Twitter performance in terms of response latency. The latency for download and upload reached 200s and

[17]http://techcrunch.com/2012/08/22/how-big-is-facebooks-data-2-5-billion-pieces-of-content-and-500-terabytes-ingested-every-day/.
[18]http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html.
[19]http://www.statisticbrain.com/twitter-statistics/.
[20]http://www.statisticbrain.com/facebook-statistics/.

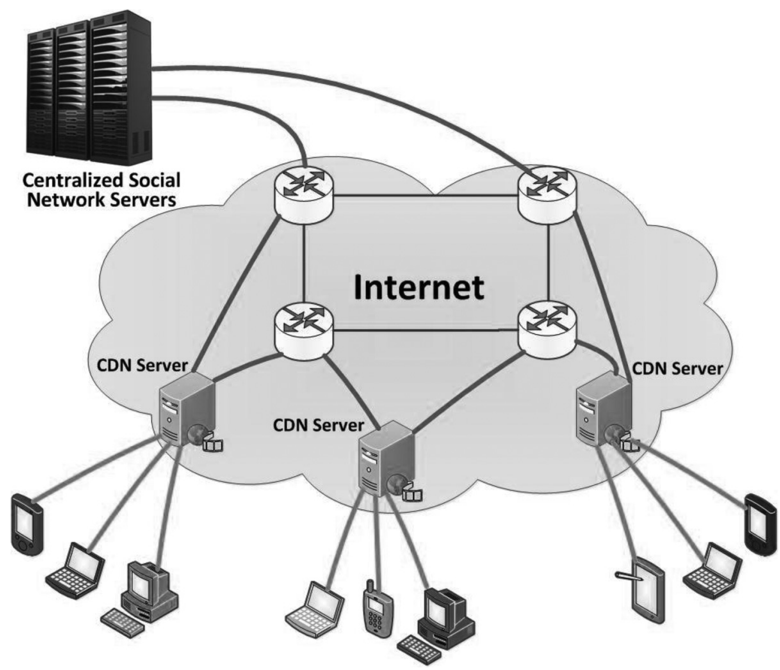

Fig. 7. Content dissemination through CDN servers.

400s during peak loads, respectively [Xu et al. 2011]. Similarly, all user requests for content are forwarded to the social network service provider and all traffic is routed to centralized servers over the Internet. User requests are still routed through the Internet even if the communicating users reside in the same Local Area Network (LAN). Routing of user requests through the large Internet paths results in higher latency [Jin et al. 2013; Wittie et al. 2010]. For instance, Facebook users outside the United States experience longer delays compared to the users residing in the United States [Jin et al. 2013; Wittie et al. 2010]. Furthermore, users that reside in rural areas are more affected by longer Internet paths due to limited available bandwidth [Johnson et al. 2012].

According to Liang et al. [2013], less frequently accessed data, called long-tailed or cold data, is stored on a single cache server within the cluster. Although CDNs can efficiently assist in the dissemination of static and frequently used contents, as illustrated in Figure 7, however, CDNs become less effective in the case of accessing dynamic and "long tailed" data [Beaver et al. 2010]. Centralized social network services providers, such as Facebook, use the CDNs to distribute the load and efficiently disseminate static content, such as photos and videos. However, CDNs cannot be used to serve dynamic content, such as user status and profile updates that change dynamically and frequently. Therefore, a tweet or profile update of a celebrity results in much more network traffic because a larger number of users have subscribed to receive the updates compared to an update from a general user [Huang and Jin 2013]. Managing elevated traffic can be a scalability challenge for centralized social networks [Huang and Jin 2013]. Moreover, occasional bursty traffic (network traffic with very high volume for very short time spans) loads are more difficult to handle as compared to consistent traffic.

An interesting solution to improve the scalability of centralized client/server-based content dissemination is proposed in Cheng and Liu [2012], who presented a novel P2P assisted video-streaming system, called *NetTube*, specifically designed for sharing short videos. Results of simulation experiments and prototype implementation reveal that NetTube significantly reduces server workload while providing improved video quality, and is scalable. Significant features of NetTube are (a) an efficient overlay that enables peers to redistribute cached videos, (b) an indexing mechanism to facilitate fast retrieval of videos, and (c) delay-aware scheduling and cluster-aware prefetching to improve playback experience and minimize delay. Architecture and implementation of NetTube is suitable for social networks due to the fact that the majority of videos shared through social networks are also short videos or photos.

Similarly, Mathieu et al. [2012] propose adopting information-centric networking for content distribution in social networks, such as Facebook, Twitter, and Google+. The authors argue that information-centric networking, due to its inherent design, presents a more natural choice for content dissemination in social networks compared with the traditional client/server paradigm or CDNs. The authors, through a case study of Twitter, show that content dissemination through information-centric networking can significantly reduce network load over the Internet as well as achieve lower latency for end users.

3.2.5. Database Scalability. Database scalability is another concern for most centralized Web-based applications [Armbrust et al. 2009]. For social networks, scalable database management is more crucial due to (a) huge amounts of data, (b) rapid development and dissemination of data and UGC, and (c) heterogeneity of content. Cloud computing has provided a paradigm that allows traditional Web servers, application servers, and social network services to scale from a few servers to thousands of high-end servers. However, the current Relational Database Management Systems (RDBMSs) are considered as “not cloud-friendly” due to the relationships and dependencies among stored data that is difficult to partition [Agrawal et al. 2011]. The current RDBMS deployments cannot be easily scaled because they do not provide support for the relevant tools and techniques required to make an RDBMS scale-out [Agrawal et al. 2011]. Moreover, traditional RDBMSs are designed to provide guaranteed consistency. As a result, RDBMSs have limited availability and scalability, especially in the case of network partition [Lakshman and Malik 2010; Lloyd et al. 2014]. In the context of social networks, traditional RDBMs are unable to provide the required latency and scalability for social networks in which data is replicated at several clusters of servers spread across different data centers that are geographically dispersed [Lakshman and Malik 2010]. For instance, Facebook used a nonrelational, distributed database, HBase, which was modelled after Google’s BigTable and written in Java. However, despite using the nonrelational database approach, the latency issues could not be alleviated due to a very high frequency of status update queries. Therefore, caching solutions are used to make the webpages display more quickly with a trade-off that the more and longer the data is cached, the less real-time it is.

Due to the scalability limitations of the traditional DBMSs mentioned earlier, large social network service providers, such as Facebook, Google+, Twitter, and Yahoo, have explored alternative data management solutions [Agrawal et al. 2011]. This resulted in development of a new class of proprietary and open-source solutions for data management called key-value stores. The main difference between the traditional RDBMS and key-value store is that the RDBMS treats the entire data stored in the database as a whole and requires the data to be consistent throughout the RDBMS [Agrawal et al. 2011]. Alternatively, key-value stores consider each key-value pair as an independent entity. Independently treating each entity leads to another advantage: data

Table IV. Comparison of Scalable Data Storage Solutions

Parameters	Cassandra	Megastore	PNUTS	Dynamo	MongoDB	COPS	SCADS
Consistency	✓*	✓	✓	✓**	✓@	✓@@	✓*
Availability	✓	✓	✓	✓	✓	✓	✓
Partition Tolerance	✓	✓	✓	✓	✓	✓	✓
Multi-Datacenter Support	✓	✓	✓	✓	✓	✓	✓
Resilience to Failures	✓	✓	✓	✓	✓	✓	✓
Rack-aware Replication	✓	NA	NA	NA	NA	NA	NA
Support for Query Language	✓	✗	✗	✗	✓	✗	✗
Open Source	✓	✗	✗	✗	✓	✗	✗
Replication Model	Asynchronous	Synchronous	Asynchronous	Asynchronous	Asynchronous	Synchronous	NA

*Eventual Consistency
**Partial Consistency
@Configurable
@@Causal Consistency

can be easily partitioned. Moreover, entities can be migrated freely from one machine to another. Recently, social network service providers, such as Facebook, Google+, and Yahoo, have successfully deployed key-value stores for their extremely large and complex applications [Agrawal et al. 2011; Lakshman and Malik 2010]. However, none of the key-value-based data storage systems, except Cassandra, provide a dynamic and rich declarative query language similar to SQL [Armbrust et al. 2009].

Here, we present some of the scalable data storage solutions currently being used by social network and cloud service providers, such as Facebook, Google, Yahoo, and Amazon, along with some solutions proposed by the research community. It is out of the scope of this survey to include a comprehensive review on scalable database solutions. Therefore, we encourage readers to see GoH13 for a more thorough survey on recent developments in database solutions. Table IV presents the comparison of selected scalable storage systems discussed later.

(1) *Cassandra:* Lakshman and Malik [2010] present a key-value-based distributed storage system, called Cassandra, initially developed at Facebook for storing structured data. Cassandra ensures high availability of user data through replicating the data across hundreds of commodity servers that are distributed across multiple data centers dispersed geographically. Instead of supporting a full relational DBMS model, Cassandra provides a simplified data model that can support customized data formats and offers dynamic control over data layouts. Some of the key features of Cassandra include scalability, failure detection and handling, partitioning, replication, and load balancing. Cassandra is being used as a backend storage solution for many Facebook services, such as Inbox search and wall comments.
(2) *Haystack:* Facebook also developed Haystack, a key-value-based data store for storing huge amounts of photos that are uploaded regularly by users [Beaver et al. 2010]. Haystack provides improved performance in terms of efficiently storing and retrieving photos. Haystack achieves high performance by reducing the need to perform an excessive number of disk lookups to retrieve metadata information. Haystack effectively reduces the size of metadata that is specifically required to store and retrieve photos. Therefore, all metadata is stored in the main memory and the lookups can be performed much faster.
(3) *Bigtable:* Google developed a scalable and distributed data storage solution for storing structured data called Bigtable [Chang et al. 2008]. Bigtable has been deployed in more than 60 Google products, such as Orkut, Google Earth, Personalized Search, and Google Analytics. Bigtable does not provide support for a full relational data

model. However, developers and clients are allowed to customize the data locality and format.

(4) *Megastore:* Google has developed another highly scalable data storage system called Megastore [Baker et al. 2011]. The system is specifically designed to provide scalability, availability, and consistency for various modern interactive online applications, such as Google+, Gmail, and GoogleTalk. Megastore combines the characteristics of scalable key-value storage with the "ease-of-use" and strong consistency of traditional DBMSs. The distinguishing feature of Megastore includes synchronous replication to ensure ACID (Atomicity, Consistency, Isolation, and Durability) properties for interactive online applications while ensuring acceptable latency.

(5) *Platform for Nimble Universal Table Storage (PNUTS):* Yahoo has developed a customized geographically distributed scalable database platform called PNUTs, which provides high parallelism for centralized Web-based applications [Cooper et al. 2008]. PNUTS has adopted asynchronous replication to achieve geographical replication and ensures low write latency for applications, such as e-mail [Cooper et al. 2008]. Moreover, PNUTS provides per-record timeline consistency that guarantees load balancing and fail-over management.

(6) *Dynamo:* Amazon developed a scalable key-value-based storage system for e-commerce applications that require high availability and reliability [DeCandia et al. 2007]. The key emphasis of Dynamo is to provide high reliability and an "always-on" experience to the diverse set of applications supported by Amazon. To achieve such an extensive availability and reliability, Dynamo has opted to sacrifice a little bit of consistency. Being a leading e-commerce service provider, Amazon supports a wide variety of applications. Therefore, Dynamo allows the developers to choose the level of consistency, availability, cost-effectiveness, and performance required by the specific applications.

(7) *MongoDB:* MongoDB is an open-source distributed database solution that belongs to the category of document stores [Grolinger et al. 2013]. MongoDB stores data in a format similar to JavaScript Object Notation (JSON) called BSON (Binary JSON). One distinguishing feature of MongoDB is that, unlike many other NoSQL databases, MongoDB supports a proprietary query language. In contrast to the traditional RDBMS, each document of a document store can have different schema. Moreover, MongoDB provides support for creating indexes on primary keys as well as other document fields for fast processing. Creation of indexes based on contents of documents differentiates document stores from key-value stores. Another significant feature of MongoDB is automatic sharding, that is, evenly balancing the load among available computing resources. However, sharding is performed on each collection separately instead of the whole database. MongoDB provides support for strong consistency that can be achieved by configuring the available options.

(8) *Cluster of Order-Preserving Servers:* Lloyd et al. [2011] have proposed Cluster of Order-Preserving Servers (COPS), a key-value-based data storage solution that provides causal consistency for clusters of servers located in the geographically distributed data centers. COPS-GT is the enhanced version, which also provides support for "get transactions." However, Lloyd et al. [2011] mentioned inefficient performance and low scalability of COPS-GT, specifically in situations such as write-heavy workloads, data center failure, and network partitions.

(9) *Scalable Consistency Adjustable Data Storage:* Armbrust et al. [2009] have presented Scalable Consistency Adjustable Data Storage (SCADS), a scalable storage system specifically designed for social computing applications. SCADS allows developers to specify consistency requirements for applications. Based on the specified requirements, SCADS intelligently determines the future resource allocation for query execution, and adaptively suggests scale-up or scale-down in the

Table V. Database Solutions Used by Social Network Service Providers

Social Network	**Database Solution**
Facebook	MySQL + Cassandra + Haystack
Google+	BigTable
Twitter	MySQL + FlockDB
LinkedIn	Voldemort + Oracle
Instagram	PostgreSQL
Pinterest	MySQL
YouTube	BigTable

cloud-computing environment. Table V presents the various database solutions used by different social network service providers.

3.3. Scalability Issues in Decentralized Social Networks

In terms of resources, such as storage space and network bandwidth, decentralized social networks are considered scalable because users can share virtually unlimited storage and bandwidth contributed by other users of a social network [Chowdhury et al. 2014]. Portable user devices, such as laptops, PDAs, and mobile phones, contribute to the overall storage and processing capacity of decentralized social networks. Increased number of users leads to increased storage capacity and enhanced bandwidth of the system [Rodrigues and Druschel 2010]. Moreover, decentralized social networks are established in limited geographical areas, such as university campuses, restaurants, and airports. Decentralized social networks can take advantage of the locality of the content, as most of the content is shared among the users located in physical proximity. Therefore, users can use the social network service without requiring any centralized server and do not require Internet connectivity. The ability of decentralized social networks to allow sharing of resources and content among the users residing in the vicinity makes decentralized social networks more scalable. However, adopting decentralized architecture for social networks gives rise to other scalability challenges, such as efficient content distribution, energy efficiency, content availability, optimal placement of UGCs, security, and privacy. In the following sections, we briefly discuss each of these challenges faced by decentralized social networks. The following scalability issues of decentralized social networks are applicable to P2P and mobile social networks.

3.3.1. Profile and Content Availability. One of the most significant challenges faced by decentralized social networks is to ensure availability of users' profiles and UGC. Due to the intermittent connectivity and ad hoc nature of decentralized social networks, ensuring continuous availability of UGC and profiles is very difficult [Narendula et al. 2010; Fu et al. 2014b]. For instance, FETHR, a decentralized microblogging social network service, does not provide any guarantee of delivery of content. Users can receive content posted when they are online, but users may not receive most of the tweets while they were offline [Xu et al. 2011]. Therefore, novel and efficient mechanisms are required that can ensure the availability of content even if users join and leave the social network at regular intervals. There are numerous solutions proposed to enhance the availability of user data in decentralized social networks [Fu et al. 2014b; Thilakarathna et al. 2013; Narendula et al. 2010]. For instance, Narendula et al. [2010] have proposed a decentralized social network named Porkut. The P2P-based social network attempts to ensure the maximum availability of user profiles and content while ensuring privacy. To achieve high availability, users' profiles and UGC are replicated on the devices of trusted friends. Access control is implemented on friends' devices to ensure privacy of stored content and profiles [Narendula et al. 2010].

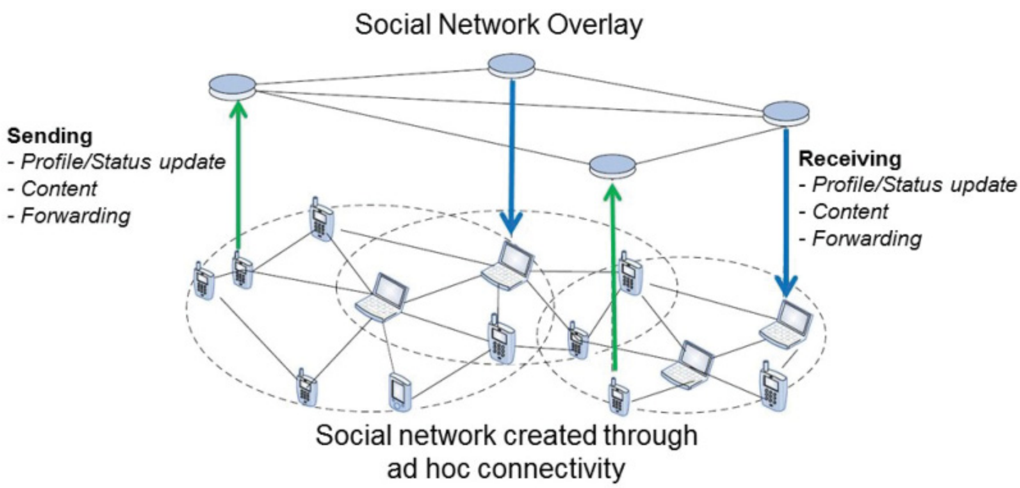

Fig. 8. Users of decentralized social network sharing content.

In decentralized social networks, UGC is stored at portable devices, such as laptops, PDAs, and mobile phones. Distributed access to UGC can raise additional scalability issues, such as optimal placement of content, availability of content when required, and efficient retrieval of content. Therefore, the optimal and scalable placement of UGC within the decentralized social network is a crucial problem that needs to be addressed [Fu et al. 2014b; Pantazopoulos et al. 2010]. The optimal placement of content in the decentralized social network is an instance of the facility location problem and, for the general case, is considered NP-hard. Therefore, novel algorithms are required for the optimal and scalable placement of UGC within the decentralized social network. Some of the proposals that address this issue include Conti et al. [2014], Chang et al. [2014], Thilakarathna et al. [2014b], Fu et al. [2014a], and Pantazopoulos et al. [2010]. For instance, Chang et al. [2014] have proposed a middleware for improving service discovery and reduce the latency in decentralized mobile social networks. Similarly, Arnaboldi et al. [2014] have also proposed a middleware that exploits users' contextual information and opportunistic wireless connectivity to efficiently discover services, resources, and users with similar interests in decentralized social networks. Another scheme is proposed by Shahriar et al. [2013] that attempts to ensure β-availability of user content. The proposed scheme exploits user online time behavior to create user groups and, consequently, replicate the data within the group to ensure β-availability. However, considering the rapid growth in the use of social networking and the massive amount of UGC and related data, such as comments and tags, highly efficient and scalable content placement, dissemination, searching, and retrieval algorithms are needed.

3.3.2. Content Distribution. The most popular feature among the users of social networks is to create and share UGC. Therefore, the most significant challenge faced by a decentralized social network is efficient delivery of content among the users of the social network. In decentralized social networks, due to the absence of any central server to facilitate content sharing, the users communicate using opportunistic and ad hoc connectivity, usually through the social network overlay, as shown in Figure 8 [Arnaboldi et al. 2014; Fu et al. 2014b]. Users sharing the content can reside in the same LAN within direct communication range of each other, or may be several hops away. The majority of the decentralized social network users connect with the social network using mobile devices, such as laptops and mobile phones. Therefore, content distribution among different mobile devices becomes more challenging due to the intermittent connectivity and limited resources [Bandara and Jayasumana 2013]. The scalability of content distribution techniques is affected by various factors, such as available bandwidth, energy consumption, freshness, content replication, and mobility of the users [Bandara and Jayasumana 2013].

To improve the data delivery ratio and reduce transmission overhead, finding the appropriate forwarding nodes is important. Selection of appropriate forwarding node is crucial for efficiently routing content requests and responses [Li et al. 2012a]. There are numerous solutions proposed in the literature to select the next forwarding nodes in decentralized systems, such as P2P networks [Kim 2014; Ferretti 2013; Mega et al. 2011] and mobile social networks [Fan et al. 2014; Li et al. 2012a]. However, the solutions developed for traditional P2P systems, such as file sharing and media streaming, cannot be directly applied to decentralized social networks due to the distinct features of social networks, such as (a) tight coupling of social networks with multiple groups, (b) frequency of interaction, and (c) support for multiple communication models [Bandara and Jayasumana 2013].

The solutions proposed for P2P networks must be extended to take advantage of the unique characteristics of decentralized social networks. For instance, Mega et al. [2011] presented a decentralized mechanism for dissemination of user profiles with the help of a social overlay over a P2P network. Similarly, Fan et al. [2014] have proposed a data dissemination scheme for mobile social networks that takes into consideration the past encounter pattern to predict future encounters. Based on the future encounter probability, the algorithm selects the best users to carry data. The users carrying data spread the data to other users using a gossiping mechanism and with the help of a gathering point. Moreover, data can be replicated at gathering points to increase the data delivery ratio. However, Mega et al. [2012] showed that, due to churn in social overlays, significant delay, up to several hours, can be observed by 1% users of a decentralized social network. Further, Mega et al. [2012] proposed a hybrid simulation/analytical framework to (a) determine an upper bound for content dissemination delay and (b) identify vital graph structures that are responsible for significant fraction of delay observed by users of large decentralized social networks. Table VI presents the comparison of some of the content distribution techniques proposed in the literature.

Similarly, in decentralized social networks, an efficient mechanism is required to notify users about profile and content updates. In contrast to centralized social networks, there is no central entity in decentralized social networks from which users can retrieve recent profile and content updates. One potential solution is to incorporate Google's PubSubHubbub (PuSH) protocol [Stirbu and Aaltonen 2014]. PuSH is a distributed publish-subscribe protocol that enables users to receive near real-time notifications about the subscribed contents [Stirbu and Aaltonen 2014]. The three main components of the PuSH protocol are (a) publishers, (b) subscribers, and (c) hubs. Publishers are the content provider while subscribers are users that are interested in receiving content from a single or multiple providers. Hubs are used to share the load of providers by replicating the newly updated contents to the designated hubs that, in turn, "push" the data to the subscribers. The advantage of hubs is two-fold. First, they provide load sharing for content providers. Second, they enable subscribers to avoid unnecessarily polling content providers asking for any updates [Kapanipathi et al. 2011]. In addition to PuSH, there are some other proposals, such as the one proposed by Olteanu and Pierre [2012] that attempts to incorporate a pubsub model for scalable content dissemination in decentralized social networks. Moreover, some of the pubsub-based solutions proposed for content delivery in P2P networks, including

Table VI. Content Dissemination Techniques for Decentralized Social Networks

Reference	History based	Encryption	Authentication	Privacy	Guaranteed delivery	Content is Replicated	Require Internet	Opportunistic connectivity
[Narendula et al. 2010]	✓	✓	✗	✓	✓	✓	✓	✗
[Thilakarathna et al. 2013]	✗	✗	✗	✓	✓	✓	✗	✓
[Conti et al. 2014]	✓	✗	✗	✓	✓	✓	✓	✗
[Fan et al. 2014]	✓	✗	✗	✗	✗	✓	✗	✓
[Mega et al. 2011]	✓	✗	✓	✓	✓	✓	✗	✓
[Shahriar et al. 2013]	✓	✗	✗	✓	✓	✓	✓	✗
[Thilakarathna et al. 2014b]	✗	✗	✗	✓	✓	✓	✓	✓
[Chang et al. 2014]	✓	✗	✗	✗	✓	✗	✓	✓
[Arnaboldi et al. 2014]	✓	✗	✗	✗	✗	✗	✗	✓

Postman [Einziger and Friedman 2014] and Vitis [Rahimian et al. 2011] can also be explored in the context of decentralized social networks.

3.3.3. Energy Efficiency. One of the most precious resources for portable devices is energy [Zhuang et al. 2010b]. Most decentralized social networks are formed by connecting portable devices that operate on batteries. Portable devices have restricted functionality due to the limited power and storage [Thilakarathna et al. 2013; Studer et al. 2011]. Moreover, in the decentralized social networks, devices are required to not only act as clients but also act as servers. Consequently, the two-way communication consumes more energy [Zhuang et al. 2010a]. Similarly, the majority of decentralized social networks leverage opportunistic connectivity for profile and content sharing [Arnaboldi et al. 2014; Fan et al. 2014; Thilakarathna et al. 2013; Mega et al. 2011]. In such scenarios, the content is shared among the participating mobile devices directly or through multihop communication using social network overlay, as illustrated in Figure 8. With the growth of a social network, the number of interactions and resulting communication also increases. Consequently, an increase in communication volume leads to an increase in energy consumption of mobile devices because mobile devices consume a significant amount of energy while scanning and communicating using WiFi, Bluetooth, or a cellular network [Banerjee et al. 2010; Castiglione et al. 2015].

Similarly, most mobile phones are equipped with different types of sensors, such as GPS, cameras, accelerometers, and light sensors. Based on the installed sensors in the mobile smartphone, various innovative context-aware and user-centric applications, such as location-based advertisement, location-based reminders, and location-based gaming, have been developed [Phan 2014]. Moreover, the users of mobile social networks benefit from location-based information, such as local events, items of interest, or discovering people with similar interests in physical proximity [Li et al. 2012a; Sarigol et al. 2010]. Location-based social applications use sensors, such as GPS, to continuously update user location and provide "always-on" connectivity [Phan 2014]. However, the sensors that are used for location sensing and provide context-aware information consume a significant amount of energy [Rachuri et al. 2014; Phan 2014; Zhuang et al. 2010b]. As the number of users participating in a decentralized social network increases, the number of interactions and frequency of accessing the underlying sensors is also increased. Whenever a user receives a request for information, such as profile and current location, the particular sensor is queried and data is transmitted over the best available channel, such as GSM, WiFi, or Bluetooth, to the intended recipient. In addition, users themselves can initiate a communication that involves accessing the sensors, for instance, by recommending a nearby restaurant or commercial offers at a particular location [Jabeur et al. 2013]. As a result of each interaction, energy consumption due to more frequent utilization of sensors and communication infrastructure may increase significantly if not carefully planned [Thilakarathna et al. 2014a]. According to Musolesi et al. [2010], sensing applications that continuously use a cellular network for Internet access can quickly consume the battery within a few hours.

The excessive use of limited battery presents a key obstacle in the wide adoption of decentralized social applications that use contextual information [Rachuri et al. 2014]. Therefore, energy efficiency is an important factor in developing solutions for decentralized social networks, such as location-based advertisements, content distribution, searching, and community detection. The proposed solution should also take into consideration the limited bandwidth availability of portable devices. However, there is a trade-off between achieving energy efficiency and performance in decentralized social networks [Kayastha et al. 2011; Banerjee et al. 2010].

There are various solutions proposed to optimize energy consumption and transmission overhead of different sensing applications that are deployed on mobile devices [Rachuri et al. 2014; Musolesi et al. 2010; Zhuang et al. 2010b; Paek et al. 2010]. For instance, Musolesi et al. [2010] have presented multiple techniques that attempt to optimize the continuous uploading of sensed data by the mobile phone. The authors analyzed the impact of the presented techniques on the energy consumption and accuracy of sensed data. The presented results indicate that significant energy savings can be obtained in terms of battery life while maintaining an acceptable level of accuracy of sensed data that does not affect application fidelity. Another sensing framework presented by Zhuang et al. [2010b] uses various techniques, such as suppression, substitution, and piggybacking of sensed data to optimize energy consumption. Similarly, Paek et al. [2010] have also presented an energy-efficient positioning system that adaptively turns on GPS to acquire user location. The proposed scheme uses timed location history to estimate user mobility and uses GPS if the estimated value exceeds a certain threshold. Rachuri et al. [2014] have presented a novel mobile sensing platform for social applications that opportunistically offloads the sensing task to the fixed infrastructure-based sensors usually installed in the smart buildings. The proposed platform can achieve up to 60% energy savings with the proposed hybrid sensing technique compared to the situation in which the mobile phone is exclusively used for sensing.

Based on this discussion, we believe that there is a need to incorporate energy efficiency while developing various user-centric and context-aware decentralized social network applications. In addition to energy-efficient mobile sensing approaches, an interesting alternative approach is proposed in Pentikousis [2010]. Pentikousis [2010] suggest modifying the whole Internet architecture from host-centric to information-centric. Moreover, the author advocates embracing a holistic approach to energy efficiency, that is, considering the entire system, rather than optimizing individual components of the system. Similarly, Fitzek et al. [2013] propose a technique that uses mobile cloud computing to reduce energy consumption and delay for sharing content among users of mobile social networks.

3.3.4. Security and Privacy. Security, privacy, and trust are the biggest challenges in decentralized social networks due to the highly distributed nature, lack of centralized coordination, and heterogeneity of the user devices [Anh et al. 2014]. Providing access to users' data while ensuring security requirements, such as confidentiality, integrity, authentication, access control, and nonrepudiation, is challenging for decentralized social networks. Moreover, in addition to simple access control, decentralized social networks need to define access control at a finer granularity, for instance, permitting specific users to read, write, and modify content. Therefore, to ensure privacy and security, most decentralized social networks store user content in encrypted format [Bonifati et al. 2010]. Another issue for decentralized social networks is how to ensure the trustworthiness of user data and content [Chen et al. 2014a; Najaflou et al. 2015].

There are several proposed systems that attempt to provide security and privacy in decentralized social networks using encryption techniques. For instance, SPAC is a P2P-based social application proposed by Bonifati et al. [2010] that supports encryption of user data. Similarly, PeerSoN is a P2P-based social network that uses encryption of users' data to enforce privacy and security [Bodriagov and Buchegger 2012]. To ensure the trustworthiness of user data, Chen et al. [2014a] have proposed a trust model that uses a cluster-based trust model that calculates the trust value of contacts based on their cluster and group affiliations. However, employing encryption leads to additional processing overhead and energy consumption, especially in the case of mobile devices having limited power, which limits the scalability of using traditional encryption techniques in decentralized social networks. Another challenge is that it is difficult to implement traditional cryptography-based encryption mechanisms due to lack of central key-distribution authority in decentralized social networks. Therefore, there is a trade-off between using encryption and performance to achieve security and privacy in decentralized social networks. In this regard, Bodriagov et al. [2014] have proposed a lightweight encryption scheme for decentralized social networks based on predicate encryption to increase the performance of the encryption scheme. The proposed scheme improves performance but reveals some information related to access policies to users having access rights.

3.3.5. Large-Scale Implementation. Due to the aforementioned scalability, to the best of our knowledge, there is no large-scale implementation of decentralized social networks. The largest implementation of a decentralized social network that has been witnessed to date is of Diaspora[21] [Thilakarathna et al. 2014a]. Diaspora is an online decentralized social networking service that allows users to connect with and use the service by selecting for themselves where their data must be stored. Users can select any server, called a *pod*, for storing user content from a large number of pods spread all over the world in the Diaspora network. To achieve better privacy and security, Diaspora allows users to create their own pod at the servers owned by users. Diaspora is an open-source platform developed using open standards, such as PubSubHubbub [Stirbu and Aaltonen 2014] and Webfinger[22]. Diaspora supports the generic features of social networking, such as sharing status updates, photos, and videos, providing users complete control over their data. Users can share data or content with anyone by organizing their contacts in multiple independent groups. However, Diaspora has not received much attention from the online social network users' community and has yet to see its full potential in the near future.

3.4. Scalability of Hybrid Social Networks

Hybrid social networks combine the features of centralized and decentralized social network architectures, and inherit strengths and weaknesses of both architectures. As noted previously, the centralized social networks require a large number of servers

[21]https://diasporafoundation.org/.
[22]http://webfinger.org/.

and infrastructure located at specially designed buildings. Centralized architecture imposes high processing load on central servers and generates large network overhead, resulting in excessive utilization of Internet bandwidth [Xu et al. 2011; Wittie et al. 2010]. Moreover, centralized servers become a performance bottleneck and can be prone to a central point of failure [Xu et al. 2011]. Similarly, the widespread adoption of decentralized social networks is challenged by other issues, such as availability, consistency, and guaranteed delivery of user content. However, there are a few proposals that suggest adopting hybrid architecture to overcome the limitations of both centralized and decentralized social networks [Stuedi et al. 2011; Xu et al. 2011; Kryczka et al. 2010; Wittie et al. 2010]. In the following, we reiterate some of the significant scalability challenges faced by both centralized and decentralized social networks. Furthermore, we discuss the characteristics of hybrid architectures that can be helpful to overcome these scalability challenges.

In hybrid social network architectures, usually a few central servers are required that can store only necessary information, such as user authentication data, content indexing, and user location. Therefore, there is no need to build large data centers that require a huge amount of energy, specially designed buildings, and dedicated network infrastructure. Similarly, in most cases, user-generated content is stored on user devices. Therefore, hybrid architectures do not require large database storage systems to store and retrieve user content. Moreover, user-generated content dissemination can be greatly improved by using hybrid schemes. For instance, content can be directly delivered to users within local networks, such as university campuses, offices, and buildings [Xu et al. 2011]. Users that are not in the range of a local network or are currently offline can fetch content through the centralized server or may request information from neighbor peers. This will not only reduce the load on central servers, but it can also improve content dissemination latency and lower unnecessary utilization of Internet bandwidth. In hybrid social networks, the required number of servers is much less compared to centralized social networks. Consequently, the impact of internal traffic, that is, intraserver communication, is very limited in hybrid architectures.

Ensuring content availability and efficient dissemination of content is one of the most significant challenges faced by decentralized social networks [Fu et al. 2014b; Fan et al. 2014; Chang et al. 2014; Fu et al. 2014a; Conti et al. 2014; Nilizadeh et al. 2012; Loupasakis et al. 2011; Mega et al. 2011]. Content availability can be improved by utilizing central servers, such as cloud storage [Thilakarathna et al. 2014, 2014b; Stuedi et al. 2011] and servers owned by service providers in which data is stored in an encrypted form. Content can be temporarily stored at central servers until fetched by all intended recipients. Similarly, content dissemination and searching in decentralized social networks can be augmented by using central indexes. For instance, Kryczka et al. [2010] proposed "uaOSN" to efficiently locate the required content and users. To ensure availability of content, use of central servers can reduce the overhead of replicating data at multiple user devices currently proposed for decentralized social networks [Narendula et al. 2010]. Another scalability challenge of decentralized social networks is energy efficiency. The major sources of energy consumption in mobile devices are sensing applications and excessive use of the Internet over cellular networks [Phan 2014; Rachuri et al. 2014; Musolesi et al. 2010; Zhuang et al. 2010b]. Hybrid mechanisms can help in improving energy efficiency by reducing Internet communication through local dissemination of the major portion of desired content [Xu et al. 2011]. Moreover, there are various solutions discussed in Section 3.3.3 to improve the energy efficiency of decentralized social networks that can be used in developing energy-efficient, location-based, and context-aware decentralized or hybrid social networks [Phan 2014; Rachuri et al. 2014; Musolesi et al. 2010; Zhuang et al. 2010b; Paek et al. 2010]. Providing effective security and authentication is more challenging in decentralized social networks. The hybrid social networks presented in Anh et al. [2014], Wang et al. [2014a], Albertini and Carminati [2014], Stuedi et al. [2011], and Kryczka et al. [2010] help in developing better security and privacy mechanisms compared to decentralized social networks. To give an overview on the workings of hybrid social networks, we present some of related work proposed in the scientific literature in the next section.

3.4.1. Related Work on Hybrid Social Networks. Kryczka et al. [2010] have proposed a user-assisted online social network "uaOSN". The uaOSN works by storing user content, such as audio, video, and photos, on the end-user devices. The social network service provider stores only the required data, such as login and profile information. In uaOSN, the service provider acts as an "indexer." All queries are directed to the indexer that, in turn, forwards the request to the appropriate hosts holding the required content. In uaOSN, user-generated content storage and the retrieval module is decentralized; remaining functionalities, such as authentication, privacy, user-profile management, social graph maintenance, and querying, remain centralized. All requests for content access are still forwarded to a centralized server that keeps track of content as well as all replicas of content spread throughout the decentralized social network. Therefore, the centralized server can limit the scalability of the social network as the number of users grows. The central server should also keep track of all users and related content, which is a difficult task, especially in the case when users are mobile.

Another hybrid architecture, called Contrail, is presented by Stuedi et al. [2011]. Contrail uses cloud infrastructure to disseminate profile updates and UGC. In Contrail, users define filters in terms of what information they are interested in receiving from their friends. For instance, users can specify wanting to be notified about the pictures uploaded by their friends where they are tagged. The filters specified by users are installed at the devices of all their friends. On the basis of specified filters, a friend's device uploads data or messages to the cloud-based application server. If the receiving user is online then data is directly transferred to the receiver without being stored at the cloud server. If the receiver is currently offline, then data is stored at the cloud-based application server, called a proxy. When the receiver comes online, the proxy transfers the data that is pending for the particular user. To implement privacy, the user devices transmit all data and filters in encrypted form. Therefore, the cloud-based application server has no visibility of the data or filters being transmitted. The major advantage of Contrail is that content dissemination is decentralized, fulfilling the criteria for privacy. However, we classified Contrail as a hybrid solution because it relies on central cloud storage servers for content dissemination. Contrail is proposed as a communication platform for social networking applications. Details regarding other social network features, such as authentication, user profiles, and social relations, are not discussed in this article. A drawback that limits the scalability of the proposed scheme is that Contrail does not support direct communication between user devices. Consequently, even if two users are in close physical proximity, they cannot directly share content.

Xu et al. [2011] proposed an architecture called Cuckoo for microblogging social network services. The authors presented a hybrid architecture that leverages the vast amount of processing and bandwidth resources of clients. Moreover, the proposed hybrid architecture also utilizes a limited number of centralized servers (server cloud) to augment the reliability and availability of content. Two content dissemination mechanisms have been proposed by the authors. First, if content is generated by a user having a smaller number of links, then the user (publisher) can directly push the content to all friends through fully decentralized unicast delivery. Second, when content is generated by a user with a very large number of links, such as celebrities or news media groups, a hybrid forwarding approach called *gossiping* is used for load-balanced delivery of content. In the gossiping mechanism, a set of interested peers, called *gossip nodes*, willingly participate in the content dissemination process to alleviate the load on the central servers. On receiving a new message from the central server, each gossip node forwards the message to a subset of directly connected peers in a decentralized manner, and the process is repeated until the message reaches a leaf node. Features such as user authentication, profile management, and social graph maintenance are centralized. Content dissemination is mostly performed using a decentralized approach with the assistance of central cloud-based servers in some cases, as mentioned earlier. The proposed hybrid architecture can simplify the management of data and greatly reduce the load on centralized servers in terms of processing and bandwidth requirements. The drawback of the proposed system is that, in the case of a gossiping mechanism, there is no guarantee that content is delivered to all followers instantly. However, peers can detect lost content based on the sequence number of received messages, and missing content can be fetched later either from the central server cloud or neighboring peers.

Wang et al. [2014a] proposed a scheme that uses mobile social networks to offload the network traffic generated by online social networks over the cellular infrastructure. Initially, a subset of users is selected to receive data from the online social network through the Internet connection over the cellular network. Later, the users that have initially received the data exploit the opportunistic connectivity, such as Wi-Fi, Bluetooth, and Device-to-Device (D2D) connectivity in Long Term Evolution (LTE) networks to share content locally using a mobile social network. The authors exploit the access time and mobility patterns of users obtained from the traces of online and mobile social networks data. In the proposed technique, the content dissemination module is decentralized while the rest of the functions of the social network remain centralized. The authors state that the proposed hybrid scheme can reduce traffic up to 85% over the cellular network while satisfying user requirements for access delay.

Wittie et al. [2010] also proposed a hybrid architecture for Facebook. The authors suggested partitioning and distributing the Facebook state across the geographically distributed servers. The authors argue that most Facebook data centers are located in the United States due to which users outside the United States experience slow response times and long delays in loading Facebook pages [Wittie et al. 2010]. The delay is caused by longer and unstable Internet paths that result in long round-trip time. The problem is further aggravated by retransmissions in the case of packet losses during transit that not only affect the user experience but also waste a lot of Internet bandwidth. The CDNs can help to improve the delivery of static content. However, a large portion of social network traffic pertains to dynamic content. Therefore, queries for dynamic content are still routed to the central Facebook data centers. The authors argue that the state of a social network can be partitioned and that carefully defined distribution can lead to significant improvement in performance without losing consistency. To overcome the aforementioned drawbacks through partitioning the social network state, the authors propose two distributed content dissemination techniques: (a) TCP proxies and (b) regional online social network caches. While both techniques can be helpful reducing transmission delay, implementation of these techniques requires more investment and deployment of additional resources at the regional level.

Famulari and Hecker [2013] propose a decentralized social network named Mantle. Mantle uses encryption and leverages cloud-based storage, such as Dropbox and Skydrive, to provide privacy and security of user data. The significant feature of Mantle is that, in addition to protecting user data and metadata, it attempts to hide the interaction and social graph of users, thereby actively preventing access from any third party. A similar approach is presented by Albertini and Carminati [2014], in which cloud-based storage is used to store user data and relationships. To achieve privacy, the data

Table VII. Comparison of Hybrid Social Networks

Reference	Application Domain	Open Source	Privacy	Control over UGC	Require Internet	Support new Applications	Encryption	Censorship	User Authentication
[Kryczka et al. 2010]	Social network	✗	✓	✓	✓	✗	✓	✓	✓
[Wittie et al. 2010]	Social Network	✗	✗	✗	✓	✗	✗	✓	✓
[Xu et al. 2011]	Microblogging	✗	✗	✗	✗	✗	✗	✗	✗
[Stuedi et al. 2011]	Social network	✗	✓	✓	✓	✗	✓	✗	✗
[Famulari and Hecker 2013]	Social network	✗	✓	✓	✓	✗	✓	✗	✓
[Mega et al. 2013]	Social network	✗	✓	✓	✓	✗	✓	✗	✗
[Anh et al. 2014]	Social network	✓	✓	✓	✓	✓	✗	✗	✗
[Wang et al. 2014a]	Social network	✗	✗	✗	✓	✗	✗	✓	✓
[Albertini and Carminati 2014]	Social network	✗	✓	✓	✓	✗	✓	✗	✓

is stored in the public cloud in encrypted form, and collaborative graph anonymization is used to obfuscate the relationships among the users within the social graph. Shared access to the data is provided based on the relationship rules specified by the owner. Another hybrid cloud-assisted content dissemination technique is proposed by Mega et al. [2013]. In the proposed technique, a profile or content update from a user is immediately uploaded to the cloud-based storage. Then, content is forwarded to other users in the social network using P2P overlay. In case user x does not receive any update from user y for a certain predefined period of time, for instance, due to transient partitioning, then user x downloads the update from cloud storage. After receiving the update from the cloud, user x will also push the update on social overlay through gossip protocol [Mega et al. 2011]. To maintain privacy, the profile and content are stored in an encrypted format on cloud storage. Other functionalities of social networks, such as authentication and social graph management, can be performed using the decentralized mechanism.

Table VII presents a comparison of some of the selected hybrid social network architectures proposed in the literature. Based on this discussion and the surveyed literature, it can be observed that hybrid architectures have many benefits compared to centralized and decentralized architecture, and can assist in solving the problems associated with scalability of social networks.

3.5. Scalability Metrics for Social Networks

Based on literature review, we have outlined certain metrics to evaluate the scalability of social network architectures. A detailed and in-depth discussion regarding eachparameter and its impact among all the discussed architectures is provided in the following sections.

3.5.1. Availability. Centralized social networks use a large number of servers and replicate user data across multiple servers to ensure the availability of content [Lakshman and Malik 2010; Traverso et al. 2015]. Hybrid architectures also provide high availability because they are supported by centralized entities, such as CDNs and cloud-based systems. However, decentralized social networks cannot guarantee the same level of availability because content is stored on user devices only and users tend to join and leave the social network at regular intervals.

3.5.2. Latency. In the case of centralized social networks, all user requests are forwarded to central data centers and traffic is routed to centralized servers over the Internet. The routing of user requests through the large Internet paths results in higher latency [Jin et al. 2013; Wittie et al. 2010]. Hybrid social networks can decrease latency by serving content from neighboring peers or nearest available servers [Jin et al. 2013; Kryczka et al. 2010]. Decentralized social networks can reduce latency in the case in which all social network users are within close physical proximity. Content can be shared among the users directly without using the Internet, leading to lower latency as well as lower burden on Internet gateways [Wittie et al. 2010]. For instance, an analysis of Facebook conducted by Wittie et al. [2010] shows that, when data is accessed from central servers of Facebook, a delay of 3.4s is observed. However, in the case that data is fetched from a regional server, the delay is 0.7s, that is, 79% less delay compared to the centralized server . These results indicate that in the case of decentralized or hybrid architectures, in which data is served from neighboring peers or regional servers, the delay can be significantly reduced.

3.5.3. Interserver Communication. A centralized social network encompasses several customized services, such as personalized advertisement, maintaining indexes, and recommendation systems that explicitly contribute to high internal network traffic. The entire business model of social networks depends on these customized services [Saez-Trumper et al. 2014]. For instance, the revenue earned by Google and Facebook for 2013 was reported to be over $50 billion and $7.8 billion, respectively [Saez-Trumper et al. 2014]. Therefore, centralized social network service providers, such as Facebook, Twitter, LinkedIn, Myspace, and Google+, hardly curtail the amount of internal traffic. Alternatively, decentralized social networks do not necessarily require a large number of servers.

3.5.4. Cost of Engineering and Resources. Centralized social networks require huge investments in terms of resources, such as buildings, servers, storage, and network equipment, as discussed in Section 3.2.2. On the other hand, decentralized social networks do not have the requirement of any centralized infrastructure. Similarly, hybrid social networks require fewer resources, such as servers, storage, and networking, at a much lower scale compared to centralized social networks.

3.5.5. Energy Consumption and Maintenance Cost. Centralized social network infrastructures consume a significant amount of energy. The components that consume major proportions of energy include servers, cooling systems, and networking equipment [Greenberg et al. 2008; Koomey et al. 2007]. In the case of decentralized social networks, energy and maintenance costs are not applicable, because there is no centralized infrastructure involved. Similarly, hybrid architectures do not require huge dedicated infrastructure. Moreover, the hybrid architectures, such as Cuckoo [Xu et al. 2011] and Contrail [Stuedi et al. 2011] use cloud infrastructure services to alleviate the cost of engineering, building, and maintaining large data centers required to support social network services.

3.5.6. Internet Bandwidth Requirement. In centralized social networks, all requests for recently uploaded dynamic data are forwarded to centralized servers, which cause a huge amount of Internet traffic routed to central social network servers [Wittie et al. 2010; Johnson et al. 2012]. For instance, according to a report published in April 2008, YouTube alone consumed as much bandwidth as was consumed by the whole Internet in 2000 [Cheng et al. 2013]. Alternatively, decentralized social networks do not require mandatory Internet connectivity because contents are shared in an ad hoc manner with opportunistic connectivity [Pantazopoulos et al. 2010]. Similarly, hybrid social networks can significantly reduce Internet bandwidth requirements because the majority of UGC can be shared locally without using the Internet [Xu et al. 2011; Kryczka et al. 2010]. According to Wittie et al. [2010], the network load on central servers of social networks, located in data centers, can be significantly reduced (up to 83% for some regions) by deploying regional servers. Deployment of regional servers not only reduces data center load but also reduces the extra burden on the Internet backbone. Wittie et al. [2010] showed that the proposed hybrid architecture consumes 91% less Internet bandwidth compared with centralized social networks.

3.5.7. Data Consistency. Centralized social networks store all user data on multiple servers that are distributed geographically [Traverso et al. 2015; Lakshman and Malik 2010; DeCandia et al. 2007]. The focus of centralized social networks is to ensure maximum availability [Lakshman and Malik 2010]. However, centralized architectures can still provide an acceptable level of consistency, because all data is managed centrally by the social network service provider [Lakshman and Malik 2010]. In the case of decentralized social networks, there is no central data server that can coordinate data storage and ensure consistency. Hybrid social networks have medium or low data consistency depending on the particular implementation.

3.5.8. Data Replication. Centralized social networks replicate user data at multiple locations to provide better availability [Traverso et al. 2015; Lakshman and Malik 2010; DeCandia et al. 2007]. In the case of failure at one location, data can be retrieved from other locations [Lakshman and Malik 2010; DeCandia et al. 2007]. In the case of decentralized social networks, if a user is offline, then content can be served from other user devices having replicated data. Similarly, in hybrid architectures, data can be replicated at user devices and central servers to enhance availability and decrease latency.

3.5.9. Privacy and Security. Despite huge success, Web-based centralized social networks face problems related to privacy and access control of user data [Jahid et al. 2012]. However, privacy and security issues do not directly affect the scalability of centralized and, in certain cases, hybrid social networks. Decentralized social networks, such as Porkut [Narendula et al. 2010] and PeerSoN [Bodriagov and Buchegger 2012] use encryption to enhance the security and privacy of user data. However, providing effective privacy and security enforced through encryption and access control is challenging and may limit the scalability of decentralized social networks, mainly due to the use of portable resource-constrained devices, such as smartphones and tablets [Castiglione et al. 2015]. In such cases, the use of cryptography for encryption becomes very challenging due to lack of central key distribution authority. Moreover, cryptography involves complex computations that heavily affect energy consumption in already resource-constrained devices [Castiglione et al. 2015]. As the number of participants in the network increases, a large list of keys needs to be maintained and distributed that can significantly increase the complexities pertaining to privacy and security. In hybrid social networks, privacy and security of user data and content depends on the data storage and dissemination scheme adopted by the particular implementation.

Table VIII summarizes the discussion related to previously discussed scalability metrics against each of the presented social network architectures: centralized, decentralized, and hybrid.

Table VIII. Scalability Metric of Social Network Architectures

Parameters	Centralized	Decentralized	Hybrid
Availability	✓	✓	✓
Latency	✓	✓	✓
Interserver Communication	✓	✗	✓
Cost of Resources	✓	✗	✓
Cost of Engineering	✓	✗	✗
Server Energy Consumption	✓	✗	✓
Infrastructure Maintenance Cost	✓	✗	✓
Internet Bandwidth Requirement	✓	✗	✓
Data Consistency	✓	✗	✓
Data Replication	✓	✓	✓
Privacy	✓	✓	✓
Security	✓	✓	✓

4. FUTURE DIRECTIONS

Although there have been a number of research initiatives in the field of social networks, the field is still in an evolutionary stage and needs significant exploration, especially in the area of privacy, user authentication, security, content availability, and interoperability. Many decentralized and hybrid social networks have been proposed in the literature. However, none of the proposed techniques provides a comprehensive solution to all the scalability issues and challenges faced by social networks due to the inherent trade-offs among various factors. In the literature survey, we have identified some interesting issues related to the field of social networks that require further research, as follows.

4.1. Interoperability Among Social Networks

Currently, all social network services, either decentralized or centralized, lack interoperability and do not allow users to share content across different social network services. Therefore, there is a need to develop a system that allows users to share media across different services platforms. For instance, “CloudSpaces” is a recent ongoing European Union project that allows users to take complete control over their data stored at “Personal Cloud” [Gracia-Tinedo et al. 2013]. The system facilitates content sharing among users while ensuring privacy. The “CloudSpaces” project aims to develop platforms that will enable interoperability between heterogeneous personal clouds. Similarly, Schwittmann et al. [2013] have proposed federated online social networks that aim to enhance user privacy and data availability. Users select the centralized servers of their choice to host data, where the data is stored in encrypted form. The data is encrypted and decrypted at user devices. Moreover, replication is used to enhance the availability of UGC. Another significant feature of the federated social network is that it allows users to interact across different social network service providers.

4.2. Service Guarantees in Decentralized Social Networks

The primary goal of a social network is to enable users to communicate and share content efficiently. Although decentralized social networks allow users to achieve the basic tasks of communicating and sharing, current decentralized social networks lack in providing guaranteed services, both in terms of availability and response time, which limits scalability and wide adoption of decentralized social networks.

4.3. P2P Overlay Management

Some researchers believe that P2P social networks do not have scalability issues [Olteanu and Pierre 2012; Rodrigues and Druschel 2010]. The reason behind such thinking is the simplified assumption that the average node degree remains the same regardless of the overall increase in the size of a P2P system [Olteanu and Pierre 2012]. On the contrary, the node degree can grow polynomially as the size of a system increases, while the effective diameter of the system decreases [Olteanu and Pierre 2012]. For instance, currently, the average number of friends each Facebook user has is more than 100; that will increase progressively [Ugander et al. 2011]. Therefore, mirroring social network links onto the P2P overlay network might become costly. A P2P social network overlay needs to minimize the number of active connections to work efficiently and remain scalable [Olteanu and Pierre 2012]. In this regard, Ferretti [2012] has proposed a scheme to restrict degree distribution in P2P overlays.

5. CONCLUSIONS

Scalability is an important parameter to determine the long-term efficiency and effectiveness of modern large-scale social networks. The task of designing and developing a robust and scalable social network becomes more challenging due to the large number of users and extremely high growth rate of user content. In this survey, we have presented an in-depth analysis of social network characteristics along with different social network architectures. Scalability issues and challenges faced by centralized, decentralized, and hybrid social networks have been discussed in detail and illustrated through appropriate examples. After thorough analysis and detailed literature review, we observe that most of the issues associated with current centralized social networks are largely due to their centralized nature and handling of users' private data and UGC. Similarly, scalability of decentralized social networks is limited by the lack of resources of portable devices, limited bandwidth, and intermittent connectivity. Therefore, use of hybrid architecture is recommended to overcome these deficiencies of centralized and decentralized social networks. Adoption of hybrid architecture for social networks can give rise to many interesting cross-layer and interdisciplinary research, such as social network analysis, security and trust, pervasive computing, and delay-tolerant networks.

REFERENCES

Divyakant Agrawal, Amr El Abbadi, Sudipto Das, and Aaron Elmore. 2011. Database scalability, elasticity, and autonomy in the cloud. In *Database Systems for Advanced Applications*. Springer, 2–15.

Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. 2013. A peek into the future: Predicting the evolution of popularity in user generated content. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. ACM, 607–616.

Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. 2008. A scalable, commodity data center network architecture. *ACM SIGCOMM Computer Communication Review* 38, 4, 63–74.

N. Al Mutawa, I. Baggili, and A. Marrington. 2012. Forensic analysis of social networking applications on mobile devices. *Digital Investigation* 9, S24–S33.

Abrar M. Alajlan and Khaled M. Elleithy. 2014. High-level abstractions in wireless sensor networks: Status, taxonomy, challenges, and future directions. In *Zone 1 Conference of the American Society for Engineering Education (ASEE Zone 1'14)*. IEEE, 1–7.

Davide Alberto Albertini and Barbara Carminati. 2014. Relationship-based information sharing in cloud-based decentralized social networks. In *Proceedings of the 4th ACM Conference on Data and Application Security and Privacy*. ACM, 297–304.

S. Androutsellis-Theotokis and D. Spinellis. 2004. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys* 36, 4, 335–371.

Dinh Tien Tuan Anh, Milind Ganjoo, Stefano Braghin, and Anwitaman Datta. 2014. Mosco: A privacy-aware middleware for mobile social computing. *Journal of Systems and Software* 92, 20–31.

Michael Armbrust, Armando Fox, David Patterson, Nick Lanham, Beth Trushkowsky, Jesse Trutna, and Haruki Oh. 2009. Scads: Scale-independent storage for social computing applications. *arXiv preprint arXiv:0909.1775*.

Valerio Arnaboldi, Marco Conti, and Franca Delmastro. 2014. CAMEO: A novel context-aware middleware for opportunistic mobile social networks. *Pervasive and Mobile Computing* 11, 148–167.

Yuval Bachar and Adam Simpkins. 2014. Introducing Wedge and FBOSS, the next steps toward a disaggregated network @ONLINE. Retrieved July 27, 2016 from https://code.facebook.com/posts/681382905244727/introducing-wedge-and-fboss-the-next-steps-toward-a-disaggregated-network/.

Yuval Bachar. 2015. Introducing 6-pack: The first open hardware modular switch @ONLINE. Retrieved July 27, 2016 from https://code.facebook.com/posts/717010588413497/introducing-6-pack-the-first-open-hardware-modular-switch/.

Lars Backstrom. 2011. Anatomy of Facebook @ONLINE. Retrieved July 27, 2016 from http://www.facebook.com/notes/facebook-data-team/anatomy-of-facebook/10150388519243859.

Jason Baker, Chris Bond, James Corbett, J. J. Furman, Andrey Khorlin, James Larson, Jean-Michel Léon, Yawei Li, Alexander Lloyd, and Vadim Yushprakh. 2011. Megastore: Providing scalable, highly available storage for interactive services. In *CIDR*, Vol. 11. 223–234.

H. M. N. Dilum Bandara and Anura P. Jayasumana. 2013. Collaborative applications over peer-to-peer systems–challenges and solutions. *Peer-to-Peer Networking and Applications* 1–20.

Nilanjan Banerjee, Sharad Agarwal, Paramvir Bahl, Ranveer Chandra, Alec Wolman, and Mark Corner. 2010. Virtual compass: Relative positioning to sense mobile social interactions. In *Pervasive Computing*. Springer, 1–21.

Partha Basuchowdhuri, Siddhartha Anand, Diksha Roy Srivastava, Khusbu Mishra, and Sanjoy Kumar Saha. 2014. Detection of communities in social networks using spanning tree. *Advanced Computing, Networking and Informatics* 2, 589–597.

Aaron Beach, Mike Gartrell, and Richard Han. 2009. Solutions to security and privacy issues in mobile social networking. In *International Conference on Computational Science and Engineering (CSE'09)*. Vol. 4. IEEE, 1036–1042.

Doug Beaver, Sanjeev Kumar, Harry C. Li, Jason Sobel, Peter Vajgel, and others. 2010. Finding a needle in haystack: Facebook's photo storage. In *OSDI*, Vol. 10. 1–8.

Paolo Bellavista, Rebecca Montanari, and Sajal K. Das. 2013. Mobile social networking middleware: A survey. *Pervasive and Mobile Computing* 9, 4, 437–453.

Theophilus Benson, Aditya Akella, and David A. Maltz. 2010. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*. ACM, 267–280.

Kashif Bilal, Samee U. Khan, Sajjad A. Madani, Khizar Hayat, Majid I Khan, Nasro Min-Allah, Joanna Kolodziej, Lizhe Wang, Sherali Zeadally, and Dan Chen. 2013. A survey on green communications using adaptive link rate. *Cluster Computing* 16, 3, 575–589.

Kashif Bilal, Saif Ur Rehman Malik, Osman Khalid, Abdul Hameed, Enrique Alvarez, Vidura Wijaysekara, Rizwana Irfan, Sarjan Shrestha, Debjyoti Dwivedy, Mazhar Ali, and others. 2014. A taxonomy and survey on green data center networks. *Future Generation Computer Systems* 36, 189–208.

Stephen Bird, Ajit Achuthan, Othman Ait Maatallah, Wenjin Hu, Kerop Janoyan, Alexis Kwasinski, Jeanna Matthews, David Mayhew, Jay Owen, and Pier Marzocca. 2014. Distributed (green) data centers: A new concept for energy, computing, and telecommunications. *Energy for Sustainable Development* 19, 83–91.

Oleksandr Bodriagov and Sonja Buchegger. 2011. Encryption for peer-to-peer social networks. In *IEEE 3rd International Conference on Privacy, Security, Risk and Trust (PASSAT'11), and 2011 IEEE 3rd International Conference on Social Computing (Socialcom)*. IEEE, 1302–1309.

Oleksandr Bodriagov and Sonja Buchegger. 2012. P2p social networks with broadcast encryption protected privacy. In *Privacy and Identity Management for Life*. Springer, 197–206.

Oleksandr Bodriagov, Gunnar Kreitz, and Sonja Buchegger. 2014. Access control in decentralized online social networks: Applying a policy-hiding cryptographic scheme and evaluating its performance. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops'14)*. IEEE, 622–628.

A. Bonifati, H. W. Wang, and R. Liu. 2010. SPac: A distributed, peer-to-peer, secure and privacy-aware social space. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 1953–1954.

Anders Brodersen, Salvatore Scellato, and Mirjam Wattenhofer. 2012. YouTube around the world: Geographic popularity of videos. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 241–250.

Nathan Bronson, Zach Amsden, George Cabrera, Prasad Chakka, Peter Dimov, Hui Ding, Jack Ferris, Anthony Giardullo, Sachin Kulkarni, Harry C. Li, and others. 2013. TAO: Facebook's distributed data store for the social graph. In *USENIX Annual Technical Conference*. 49–60.

Nathan Bronson, Thomas Lento, and Janet L. Wiener. 2015. Open data challenges at Facebook. In *IEEE 31st International Conference on Data Engineering (ICDE'15)*. IEEE, 1516–1519.

Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros. 2010. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Algorithms and Architectures for Parallel Processing*. Springer, 13–31.

Claudia Canali, Michele Colajanni, and Riccardo Lancellotti. 2011. Data acquisition in social networks: Issues and proposals. In *Proceedings of the International Workshop on Services and Open Sources (SOS'11), Brest, France*.

Arcangelo Castiglione, Francesco Palmieri, Ugo Fiore, Aniello Castiglione, and Alfredo De Santis. 2015. Modeling energy-efficient secure communications in multi-mode wireless mobile devices. *Journal of Computer and System Sciences* 81, 8, 1464–1478.

Srimat T. Chakradhar and Anand Raghunathan. 2010. Best-effort computing: Re-thinking parallel software and hardware. In *Proceedings of the 47th Design Automation Conference*. ACM, 865–870.

Chii Chang, Satish Narayana Srirama, and Sea Ling. 2014. Towards an adaptive mediation framework for mobile social network in proximity. *Pervasive and Mobile Computing* 12, 179–196.

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. 2008. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems* 26, 2, 4.

Kyle Chard, Simon Caton, Omer Rana, and Daniel S. Katz. 2012. A social content delivery network for scientific cooperation: Vision, design, and architecture. In *SC Companion: High Performance Computing, Networking, Storage and Analysis (SCC'12)*. IEEE, 1058–1067.

Shuhong Chen, Guojun Wang, and Weijia Jia. 2014a. Cluster-group based trusted computing for mobile social networks using implicit social behavioral graph. *Future Generation Computer Systems.*

Shuang Chen, Yanzhi Wang, and Massoud Pedram. 2014b. Concurrent placement, capacity provisioning, and request flow control for a distributed cloud infrastructure. In *Design, Automation and Test in Europe Conference and Exhibition (DATE'14)*. IEEE, 1–6.

Xu Cheng and Jiangchuan Liu. 2012. Exploring interest correlation for peer-to-peer socialized video sharing. *ACM Transactions on Multimedia Computing, Communications, and Applications* 8, 1, 5.

Xu Cheng, Jiangchuan Liu, and Cameron Dale. 2013. Understanding the characteristics of Internet short video sharing: A YouTube-based measurement study. *IEEE Transactions on Multimedia* 15, 5, 1184–1194.

Shihabur Rahman Chowdhury, Arup Raton Roy, Maheen Shaikh, and Khuzaima Daudjee. 2014. A taxonomy of decentralized online social networks. *Peer-to-Peer Networking and Applications* 8, 3, 367–383.

Marco Conti, Andrea De Salve, Barbara Guidi, Francesco Pitto, and Laura Ricci. 2014. Trusted dynamic storage for Dunbar-based P2P online social networks. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences*. Springer, 400–417.

Brian F. Cooper, Raghu Ramakrishnan, Utkarsh Srivastava, Adam Silberstein, Philip Bohannon, Hans-Arno Jacobsen, Nick Puz, Daniel Weaver, and Ramana Yerneni. 2008. PNUTS: Yahoo!'s hosted data serving platform. *Proceedings of the VLDB Endowment* 1, 2, 1277–1288.

Yanqing Cui and Mikko Honkala. 2011. The consumption of integrated social networking services on mobile devices. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 53–62.

Cristiano André da Costa, Dante Zaupa, Jorge Luis Victoria Barbosa, Rodrigo da Rosa Righi, João de Camargo, Nikolas Daroit, and Adenauer Correa Yamin. 2014. A spontaneous social network based on mobile devices. *Social Network Analysis and Mining* 4, 1, 1–13.

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. 2007. Dynamo: Amazon's highly available key-value store. In *SOSP*, Vol. 7. 205–220.

Manlio Del Giudice, Maria Rosaria Della Peruta, and Elias G. Carayannis. 2014. Social websites: Practices and tools for emerging markets. In *Social Media and Emerging Economies*. Springer, 7–31.

Wei Deng, Fangming Liu, Hai Jin, Bo Li, and Dan Li. 2014. Harnessing renewable energy in cloud datacenters: Opportunities and challenges. *IEEE Network,* 28, 1, 48–55.

Khosrow Ebrahimi, Gerard F. Jones, and Amy S. Fleischer. 2014. A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities. *Renewable and Sustainable Energy Reviews* 31, 622–638.

Gil Einziger and Roy Friedman. 2014. Postman: An elastic highly resilient publish/subscribe framework for self sustained service independent P2P networks. In *Stabilization, Safety, and Security of Distributed Systems*. Springer, 78–92.

Nicole B. Ellison and others. 2007. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication* 13, 1, 210–230.

Antonino Famulari and Artur Hecker. 2013. Mantle: A novel DOSN leveraging free storage and local software. In *Advanced Infocomm Technology*. Springer, 213–224.

Bo Fan, Supeng Leng, Kun Yang, and Qiang Liu. 2014. GPS: A method for data sharing in mobile social networks. In *2014 IFIP Networking Conference*. IEEE, 1–9.

Nathan Farrington and Alexey Andreyev. 2013. Facebook's data center network architecture. In *IEEE Optical Interconnects Conference*. Citeseer.

Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2011. Helios: A hybrid electrical/optical switch architecture for modular data centers. *ACM SIGCOMM Computer Communication Review* 41, 4, 339–350.

Stefano Ferretti. 2012. On the degree distribution of faulty peer-to-peer overlay networks. *EAI Endorsed Transactions on Complex Systems* 12, 1. DOI: http://dx.doi.org/10.4108/trans.cs.2012.10-12.e2

Stefano Ferretti. 2013. Gossiping for resource discovering: An analysis based on complex network theory. *Future Generation Computer Systems* 29, 6, 1631–1644.

Frank H. P. Fitzek, Janus Heide, Morten Videbæk Pedersen, and Marcos Katz. 2013. Implementation of network coding for social mobile clouds [applications corner]. *IEEE Signal Processing Magazine* 30, 1, 159–164.

Songling Fu, Ligang He, Xiangke Liao, Chenlin Huang, Cheng Chang, and Bo Gao. 2014a. Cadros: The cloud-assisted data replication in decentralized online social networks. In *IEEE International Conference on Services Computing (SCC'14)*. IEEE, 43–50.

Songling Fu, Ligang He, Xiangke Liao, Kenli Li, and Chenlin Huang. 2014b. Analyzing the impact of storage shortage on data availability in decentralized online social networks. *The Scientific World Journal*.

Sabrina Gaito, Matteo Zignani, Gian Paolo Rossi, Alessandra Sala, Xiaohan Zhao, Haitao Zheng, and Ben Y. Zhao. 2012. On the bursty evolution of online social networks. In *Proceedings of the 1st ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research*. ACM, 1–8.

Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. 2012. A framework for ranking of cloud computing services. *Future Generation Computer Systems*.

Chaima Ghribi. 2014. *Energy Efficient Resource Allocation in Cloud Computing Environments*. Ph.D. Dissertation. Institut National des Télécommunications, Ile-de-France, France..

Raúl Gracia-Tinedo, Marc Sanchez Artigas, Adrian Moreno-Martinez, Cristian Cotes, and Pedro Garcia Lopez. 2013. Actively measuring personal cloud storage. In *IEEE 6th International Conference on Cloud Computing (CLOUD'13)*. IEEE, 301–308.

Kalman Graffi, Christian Gross, Dominik Stingl, Daniel Hartung, Aleksandra Kovacevic, and Ralf Steinmetz. 2011. LifeSocial. KOM: A secure and P2P-based solution for online social networks. In *IEEE Consumer Communications and Networking Conference (CCNC'11)*. IEEE, 554–558.

Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. 2008. The cost of a cloud: Research problems in data center networks. *ACM SIGCOMM Computer Communication Review* 39, 1, 68–73.

Albert Greenberg, James R. Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta. 2011. VL2: A scalable and flexible data center network. *Communications of the ACM* 54, 3, 95–104.

Katarina Grolinger, Wilson A. Higashino, Abhinav Tiwari, and Miriam A. M. Capretz. 2013. Data management in cloud environments: NoSQL and NewSQL data stores. *Journal of Cloud Computing: Advances, Systems and Applications* 2, 1, 22.

Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. 2009. BCube: A high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Computer Communication Review* 39, 4, 63–74.

Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu. 2008. Dcell: A scalable and fault-tolerant network structure for data centers. *ACM SIGCOMM Computer Communication Review* 38, 4, 75–86.

Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. 2011. Augmenting data center networks with multi-gigabit wireless links. In *ACM SIGCOMM Computer Communication Review*, Vol. 41. ACM, 38–49.

Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R. Das, Jon P. Longtin, Himanshu Shah, and Ashish Tanwer. 2015. FireFly: A reconfigurable wireless data center fabric using free-space optics. *ACM SIGCOMM Computer Communication Review* 44, 4, 319–330.

Ali Hammadi and Lotfi Mhamdi. 2014. A survey on architectures and energy efficiency in data center networks. *Computer Communications* 40, 1–21.

Abdelbaset S. Hamza, Jitender S. Deogun, and Dennis R. Alexander. 2014. Free space optical data center architecture design with fully connected racks. In *IEEE Global Communications Conference (GLOBECOM'14)*. IEEE, 2192–2197.

Abdelbaset S. Hamza, Jitender S. Deogun, and Dennis R. Alexander. 2015. Evolution of data centers: A critical analysis of standards and challenges for FSO links. In *IEEE Conference on Standards for Communications and Networking (CSCN'15)*. IEEE, 100–105.

Jinzhou Huang and Hai Jin. 2013. SoMed: A hybrid DHT framework towards scalable decentralized microblogging services. In *International Conference on Parallel and Distributed Systems (ICPADS'13)*. IEEE, 534–539.

Nafaâ Jabeur, Sherali Zeadally, and Biju Sayed. 2013. Mobile social networking applications. *Communications of the ACM* 56, 3, 71–79.

Sonia Jahid, Shirin Nilizadeh, Payal Mittal, Nikita Borisov, and Apu Kapadia. 2012. DECENT: A decentralized architecture for enforcing privacy in online social networks. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops'12)*. IEEE, 326–332.

Long Jin, Yang Chen, Tianyi Wang, Pan Hui, and Athanasios V. Vasilakos. 2013. Understanding user behavior in online social networks: A survey. *IEEE Communications Magazine* 51, 9, 144–150.

David L. Johnson, Elizabeth M. Belding, and Gertjan Van Stam. 2012. Network traffic locality in a rural African village. In *Proceedings of the 5th International Conference on Information and Communication Technologies and Development*. ACM, 268–277.

Kailash Joshi. 2013. *Graph Visualization Using the NoSQL Database*. Ph.D. dissertation. North Dakota State University, Fargo, ND.

P. Kapanipathi, J. Anaya, A. Sheth, B. Slatkin, and A. Passant. 2011. Privacy-aware and scalable content dissemination in distributed social networks. *The Semantic Web–ISWC 2011* 157–172.

Nipendra Kayastha, Dusit Niyato, Ping Wang, and Ekram Hossain. 2011. Applications, architectures, and protocol design issues for mobile social networks: A survey. *Proceedings of IEEE* 99, 12, 2130–2158.

Seokhoon Kim. 2014. QoS-aware data forwarding architecture for multimedia streaming services in hybrid peer-to-peer networks. *Peer-to-Peer Networking and Applications* 1–10.

W. Kim, O. R. Jeong, and S. W. Lee. 2010. On social web sites. *Information Systems* 35, 2, 215–236.

Dzmitry Kliazovich, Pascal Bouvry, and Samee Ullah Khan. 2013. DENS: Data center energy-efficient network-aware scheduling. *Cluster Computing* 16, 1, 65–75.

David Koll, Jun Li, and Xiaoming Fu. 2014. SOUP: An online social network by the people, for the people. In *Proceedings of the 15th International Middleware Conference*. ACM, 193–204.

Jonathan Koomey, Kenneth Brill, Pitt Turner, John Stanley, and Bruce Taylor. 2007. A simple model for determining true total cost of ownership for data centers. Uptime Institute White Paper, Version 2.

Nicolas Kourtellis, Joshua Finnis, Paul Anderson, Jeremy Blackburn, Cristian Borcea, and Adriana Iamnitchi. 2010. Prometheus: User-controlled P2P social data management for socially-aware applications. In *Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware*. Springer-Verlag, 212–231.

Michal Kryczka, Ruben Cuevas, Carmen Guerrero, Eiko Yoneki, and Arturo Azcorra. 2010. A first step towards user assisted online social networks. In *Proceedings of the 3rd Workshop on Social Network Systems*. ACM, 6.

Sanjeev Kumar. 2012. Analyzing the Facebook workload. In *IEEE International Symposium on Workload Characterization (IISWC'12)*. IEEE, 111–112.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 591–600.

Avinash Lakshman and Prashant Malik. 2010. Cassandra: A decentralized structured storage system. *ACM SIGOPS Operating Systems Review* 44, 2, 35–40.

Hemank Lamba, Momin M. Malik, Constantine Nakos, and Jurgen Pfeffer. 2015. Rich people don't have more followers! Overcoming social inequality with social media. In *Proceedings of the International Social Computing, Behavioral-Cultural Modeling and Prediction Conference (SBP'15) Grand Data Challenge*, Vol. 2015.

Haitao Li, Haiyang Wang, Jiangchuan Liu, and Ke Xu. 2012b. Video sharing in online social networks: Measurement and analysis. In *Proceedings of the 22nd International Workshop on Network and Operating System Support for Digital Audio and Video*. ACM, 83–88.

J. Li, H. Wang, and S. U. Khan. 2012a. A semantics-based approach to large-scale mobile social networking. *Mobile Networks and Applications* 17, 2, 192–205.

John Liang, James Luo, Mark Drayton, Rajesh Nishtala, Richard Liu, Nick Hammer, Jason Taylor, and Bill Jia. 2013. Storage and performance optimization of long tail key access in a social network. In *Proceedings of the 3rd International Workshop on Cloud Data and Platforms*. ACM, 1–6.

Wyatt Lloyd, Michael J. Freedman, Michael Kaminsky, and David G. Andersen. 2011. Don't settle for eventual: Scalable causal consistency for wide-area storage with COPS. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles*. ACM, 401–416.

Wyatt Lloyd, Michael J. Freedman, Michael Kaminsky, and David G. Andersen. 2014. Don't settle for eventual consistency. *Communications of the ACM* 57, 5, 61–68.

Andreas Loupasakis, Nikos Ntarmos, Peter Triantafillou, and Darko Makreshanski. 2011. eXO: Decentralized autonomous scalable social networking. In *Proceedings of CIDR*.

R. Lubke, Daniel Schuster, and Alexander Schill. 2011. Mobilisgroups: Location-based group formation in mobile social networks. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops'11)*. IEEE, 502–507.

Joaquim Marques and Carlos Serrão. 2014. Improving user content privacy on social networks using rights management systems. *Annals of Telecommunications-Annales Des Télécommunications* 69, 1–2, 37–45.

Bertrand Mathieu, Patrick Truong, Wei You, and Jean-François Peltier. 2012. Information-centric networking: A natural design for social network applications. *IEEE Communications Magazine* 50, 7, 44–51.

Giuliano Mega, Alberto Montresor, and Gian Pietro Picco. 2011. Efficient dissemination in decentralized social networks. In *IEEE International Conference on Peer-to-Peer Computing (P2P'11)*. IEEE, 338–347.

Giuliano Mega, Alberto Montresor, and Gian Pietro Picco. 2012. On churn and communication delays in social overlays. In *IEEE 12th International Conference on Peer-to-Peer Computing (P2P'12)*. IEEE, 214–224.

Giuliano Mega, Alberto Montresor, and Gian Pietro Picco. 2013. Cloud-assisted dissemination in social overlays. In *2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P'13)*. IEEE, 1–5.

Aravind Menon. 2012. Big data@ Facebook. In *Proceedings of the 2012 Workshop on Management of Big Data Systems*. ACM, 31–32.

Rich Miller. 2013. Facebooks Power Footprint Growing, Moving East. Retrieved July 27, 2016 from http://www.datacenterknowledge.com/archives/2013/07/22/facebooks-shifting-power-footprint/.

Mirco Musolesi, Mattia Piraccini, Kristof Fodor, Antonio Corradi, and Andrew T. Campbell. 2010. Supporting Energy-Efficient Uploading Strategies for Continuous Sensing Applications on Mobile Phones. In *Proceedings of the 8th International Conference on Pervasive Computing*, Helsinki, Finland, May 17–20, 2010. Springer Berlin Heidelberg, Berlin, Heidelberg, 355–372. DOI:http://dx.doi.org/10.1007/978-3-642-12654-3_21

Yashar Najaflou, Behrouz Jedari, Feng Xia, Laurence T. Yang, and Mohammad S. Obaidat. 2015. Safety challenges and solutions in mobile social networks. *IEEE Systems Journal* 9, 3 (2015), 834–854.

Rammohan Narendula, Thanasis G. Papaioannou, and Karl Aberer. 2010. Privacy-aware and highly-available OSN profiles. In *19th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'10)*. IEEE, 211–216.

Rammohan Narendula, Thanasis G. Papaioannou, and Karl Aberer. 2012. A decentralized online social network with efficient user-driven replication. In *International Conference on Privacy, Security, Risk and Trust (PASSAT'12), and 2012 International Conference on Social Computing (SocialCom'12)*. IEEE, 166–175.

Raj Nepali and Yong Wang. 2014. SocBridge: Bridging the gap between online social networks. In *Twentieth Americas Conference on Information Systems*. Savannah, 1–7.

A. T. Nguyen, B. Li, M. Welzl, and F. Eliassen. 2011. Stir: Spontaneous social peer-to-peer streaming. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS'11)*. IEEE, 816–821.

Shirin Nilizadeh, Sonia Jahid, Prateek Mittal, Nikita Borisov, and Apu Kapadia. 2012. Cachet: A decentralized architecture for privacy preserving social networking with caching. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*. ACM, 337–348.

Rajesh Nishtala, Hans Fugal, Steven Grimm, Marc Kwiatkowski, Herman Lee, Harry C. Li, Ryan McElroy, Mike Paleczny, Daniel Peek, Paul Saab, and others. 2013. Scaling memcache at Facebook. In *Presented as Part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI'13)*. 385–398.

Alexandra Olteanu and Guillaume Pierre. 2012. Towards robust and scalable peer-to-peer social networks. In *Proceedings of the 5th Workshop on Social Network Systems (SNS'12)*. ACM, New York, NY, Article 10, 6 pages. DOI:http://dx.doi.org/10.1145/2181176.2181186

John Ousterhout, Parag Agrawal, David Erickson, Christos Kozyrakis, Jacob Leverich, David Mazières, Subhasish Mitra, Aravind Narayanan, Diego Ongaro, Guru Parulkar, and others. 2011. The case for RAMCloud. *Communications of the ACM* 54, 7, 121–130.

Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. 2010. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. ACM, 299–314.

George Pallis, Demetrios Zeinalipour-Yazti, and Marios D. Dikaiakos. 2011. Online social networks: Status and trends. In *New Directions in Web Data Management 1*. Springer, 213–234.

Panagiotis Pantazopoulos, Ioannis Stavrakakis, Andrea Passarella, and Marco Conti. 2010. Efficient social-aware content placement in opportunistic networks. In *7th International Conference on Wireless On-demand Network Systems and Services (WONS'10)*. IEEE, 17–24.

Thomas Paul, Antonino Famulari, and Thorsten Strufe. 2014. A survey on decentralized online social networks. *Computer Networks* 75, 437–452.

Kostas Pentikousis. 2010. In search of energy-efficient mobile networking. *Communications Magazine, IEEE* 48, 1, 95–103.

Thomas Phan. 2014. Intelligent energy-efficient triggering of geolocation fix acquisitions based on transitions between activity recognition states. In *Mobile Computing, Applications, and Services*. Springer, 104–121.

Josep M. Pujol, Vijay Erramilli, Georgos Siganos, Xiaoyuan Yang, Nikos Laoutaris, Parminder Chhabra, and Pablo Rodriguez. 2011. The little engine (s) that could: Scaling online social networks. *ACM SIGCOMM Computer Communication Review* 41, 4, 375–386.

Kiran K. Rachuri, Christos Efstratiou, Ilias Leontiadis, Cecilia Mascolo, and Peter J. Rentfrow. 2014. Smartphone sensing offloading for efficiently supporting social sensing applications. *Pervasive and Mobile Computing* 10, 3–21.

Fatemeh Rahimian, Sarunas Girdzijauskas, Amir H. Payberah, and Seif Haridi. 2011. Vitis: A gossip-based hybrid overlay for Internet-scale publish/subscribe enabling rendezvous routing in unstructured overlay networks. In *IEEE International Parallel & Distributed Processing Symposium (IPDPS'11),* IEEE, 746–757.

Fatemeh Raji, Mohammad Davarpanah Jazi, and Ali Miri. 2014. PESCA: A peer-to-peer social network architecture with privacy-enabled social communication and data availability. *IET Information Security* 9, 1, 73–80.

Nabeel A. Riza and Philip J. Marraccini. 2012. Power smart in-door optical wireless link applications. In *8th International Wireless Communications and Mobile Computing Conference (IWCMC'12)*. IEEE, 327–332.

R. Rodrigues and P. Druschel. 2010. Peer-to-peer systems. *Communications of the ACM* 53, 10, 72–82.

Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C. Snoeren. 2015. Inside the social network's (datacenter) network. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. ACM, 123–137.

Diego Saez-Trumper, Yabing Liu, Ricardo Baeza-Yates, Balachander Krishnamurthy, and Alan Mislove. 2014. Beyond CPM and CPC: Determining the value of users on OSNs. In *Proceedings of the 2nd Edition of the ACM Conference on Online Social Networks*. ACM, 161–168.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 851–860.

E. Sarigol, O. Riva, and G. Alonso. 2010. A tuple space for social networking on mobile phones. In *IEEE 26th International Conference on Data Engineering (ICDE'10)*. IEEE, 988–991.

Lorenz Schwittmann, Christopher Boelmann, Matthaus Wander, and Torben Weis. 2013. SoNet–Privacy and replication in federated online social networks. In *IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW'13)*. IEEE, 51–57.

Nashid Shahriar, Shubhajit Roy Chowdhury, Mahfuza Sharmin, Rizwan Ahmed, Raouf Boutaba, and Bertrand Mathieu. 2013. Ensuring beta-availability in P2P social networks. In *IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW'13)*. IEEE, 150–155.

Rajesh Sharma and Anwitaman Datta. 2012. Supernova: Super-peers based architecture for decentralized online social networks. In *4th International Conference on Communication Systems and Networks (COMSNETS'12)*. IEEE, 1–10.

Ji-Yong Shin, Emin Gn Sirer, Hakim Weatherspoon, and Darko Kirovski. 2013. On the feasibility of completely wirelesss datacenters. *IEEE/ACM Transactions on Networking* 21, 5, 1666–1679.

Vlad Stirbu and Timo Aaltonen. 2014. Enabling real-time resource oriented architectures with rest observers. In *REST: Advanced Research Topics and Practical Applications*. Springer, 51–67.

Patrick Stuedi, Iqbal Mohomed, Mahesh Balakrishnan, Z. Morley Mao, Venugopalan Ramasubramanian, Doug Terry, and Ted Wobber. 2011. Contrail: Enabling decentralized social networks on smartphones. In *Middleware 2011*. Springer, 41–60.

Panagiotis Symeonidis, Dimitrios Ntempos, and Yannis Manolopoulos. 2014. Online social networks. In *Recommender Systems for Location-based Social Networks*. Springer, 21–34.

Kanchana Thilakarathna, Xinlong Guan, and Aruna Seneviratne. 2014. Demo: Yalut–user-centric social networking overlay. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 360–361.

Kanchana Thilakarathna, Abd Karim, Abdul Alim, Henrik Petander, and Aruna Seneviratne. 2013. Mobitribe: Enabling device centric social networking on smart mobile devices. In *10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'13)*. IEEE, 230–232.

Kanchana Thilakarathna, Henrik Petander, Julián Mestre, and Aruna Seneviratne. 2014a. Mobitribe: Cost efficient distributed user generated content sharing on smartphones. *IEEE Transactions on Mobile Computing* 13, 9, 2058–2070.

Kanchana Thilakarathna, Aruna Seneviratne, Aline Carneiro Viana, and Henrik Petander. 2014b. User generated content dissemination in mobile social networks through infrastructure supported content replication. *Pervasive and Mobile Computing* 11, 132–147.

Kanchana Thilakarathna, Aline Carneiro Viana, Aruna Seneviratne, and Henrik Petander. 2013. Mobile social networking through friend-to-friend opportunistic content dissemination. In *Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 263–266.

Ashish Thusoo, Zheng Shao, Suresh Anthony, Dhruba Borthakur, Namit Jain, Joydeep Sen Sarma, Raghotham Murthy, and Hao Liu. 2010. Data warehousing and analytics infrastructure at Facebook. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. ACM, 1013–1020.

Stefano Traverso, Kevin Huguenin, Ionut Trestian, Vijay Erramilli, Nikolaos Laoutaris, and Konstantina Papagiannaki. 2015. Social-aware replication in geo-diverse online systems. *IEEE Transactions on Parallel and Distributed Systems* 26, 2, 584–593.

J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. 2011. The anatomy of the Facebook social graph. *arXiv preprint arXiv:1111.4503* (2011).

Guohui Wang, David G. Andersen, Michael Kaminsky, Konstantina Papagiannaki, T. S. Ng, Michael Kozuch, and Michael Ryan. 2011. c-through: Part-time optics in data centers. *ACM SIGCOMM Computer Communication Review* 41, 4, 327–338.

Lizhe Wang and Samee U. Khan. 2013. Review of performance metrics for green data centers: A taxonomy study. *The Journal of Supercomputing* 1–18.

Lin Wang, Fa Zhang, Jordi Arjona Aroca, Athanasios V. Vasilakos, Kai Zheng, Chenying Hou, Dan Li, and Zhiyong Liu. 2014c. GreenDCN: A general framework for achieving energy efficiency in data center networks. *IEEE Journal on Selected Areas in Communications* 32, 1, 4–15.

Xiaofei Wang, Min Chen, Zhu Han, Dapeng Oliver Wu, and Ted Taekyoung Kwon. 2014a. TOSS: Traffic offloading by social network service-based opportunistic sharing in mobile social networks. In *Proceedings of IEEE INFOCOM*. IEEE, 2346–2354.

Xiaoying Wang, Xiaojing Liu, Lihua Fan, and Jianqiang Huang. 2014b. Energy-aware resource management and green energy use for large-scale datacenters: A survey. In *Proceedings of International Conference on Computer Science and Information Technology*. Springer, 555–563.

Mike P. Wittie, Veljko Pejovic, Lara Deek, Kevin C. Almeroth, and Ben Y. Zhao. 2010. Exploiting locality of interest in online social networks. In *Proceedings of the 6th International Conference*. ACM, 25.

Tianyin Xu, Yang Chen, Lei Jiao, Ben Y. Zhao, Pan Hui, and Xiaoming Fu. 2011. Scaling microblogging services with divergent traffic demands. In *Proceedings of the 12th International Middleware Conference*. International Federation for Information Processing, 20–39.

Jingqi Yang, Chuanchang Liu, Yanlei Shang, Bo Cheng, Zexiang Mao, Chunhong Liu, Lisha Niu, and Junliang Chen. 2014. A cost-aware auto-scaling approach using the workload prediction in service clouds. *Information Systems Frontiers* 16, 1, 7–18.

Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li. 2010. E-smalltalker: A distributed mobile system for social networking in physical proximity. In *IEEE 30th International Conference on Distributed Computing Systems (ICDCS'10)*. IEEE, 468–477.

Mindi Yuan, David Stein, Berenice Carrasco, Joana M. F. Trindade, and Yi Lu. 2012. Partitioning social networks for fast retrieval of time-dependent queries. In *IEEE 28th International Conference on Data Engineering Workshops (ICDEW'12)*. IEEE, 205–212.

Bin Zhou, Jian Pei, and WoShun Luk. 2008. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM SIGKDD Explorations Newsletter* 10, 2, 12–22.

Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y. Zhao, and Haitao Zheng. 2012. Mirror mirror on the ceiling: Flexible wireless links for data centers. *ACM SIGCOMM Computer Communication Review* 42, 4, 443–454.

Zhenyun Zhuang, Sandeep Kakumanu, Yeonsik Jeong, Raghupathy Sivakumar, and Aravind Velayutham. 2010a. Mobile hosts participating in peer-to-peer data networks: Challenges and solutions. *Wireless Networks* 16, 8 (2010), 2313–2333.

Zhenyun Zhuang, Kyu-Han Kim, and Jatinder Pal Singh. 2010b. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. ACM, 315–330.

Received November 2015; revised May 2016; accepted June 2016