hhu Heinrich Heine
Universität
Düsseldorf

Peer-to-Peer Mechanisms for Fully Decentralized, Secure and Scalable Online Social Networks

Inaugural-Dissertation

for the attainment of the title of doctor
in the Faculty of Mathematics and Natural Sciences
at the Heinrich Heine University Düsseldorf

presented by

Newton Wafula Masinde
from Nakuru, Kenya

Düsseldorf, December 2020

from the institute of Computer Science
at the Heinrich Heine University Düsseldorf

Published by permission of the
Faculty of Mathematics and Natural Sciences at
Heinrich Heine University Düsseldorf

Supervisor: Dr.-Ing. Kalman Graffi
Co-supervisor: Prof. Dr. Stefan Conrad

Date of the oral examination: 27/01/2021

Für Lynne...

Abstract

Social media usage in the twenty-first century has infiltrated many facets of everyday life. Depending on the users' needs, different social media platforms are available, such as social networks like Facebook and LinkedIn, blogs like Tumblr and WordPress, microblogs like Twitter, and media sharing networks like YouTube. Over the past decade, social networks have received much attention due to reliance on a centralized computing platform. We summarize the concerns into two categories, accumulated costs due to centralization and security and privacy concerns, which arise because of a single provider controlling and owning all the data uploaded. The need to offset the costs results in user data monetization, leading to privacy concerns. The prevailing proposal is to move to a decentralized computing model, which we anticipate will address the problems. The federated network is a break from the centralized model, with services provided by several independent providers. Federation also enables the network to become resilient to censorship and significantly reduces the costs incurred due to centralization but does not eliminate it. Thus, even with the federated networks, monetization of user data can occur, but not to the same extent as the centralized online social networks (OSNs).

Therefore, the alternative is to fully decentralize the computing platform and use the peer-to-peer (P2P) model to address both concerns. It transfers the infrastructural costs from providers to the users who, now, directly provide the infrastructure to keep the network running. It also ensures the users' data is private and stored on the users' devices. Besides this, it ensures that the network remains resilient to censorship. However, using P2P technology comes with the challenge of developing mechanisms that improve service delivery to match that offered by centralized OSNs.

To motivate P2P technology for implementing decentralized OSNs, we review social networks in general to discover the user requirements and the functional and non-functional system requirements. We derive the technical aspects from these requirements into a four-fold architecture composed of the overlay, the core framework, social network elements, and the graphical user interface. The overlay offers support for address management, routing, and security. Storage, communication, searching mechanisms, access control mechanisms, and monitoring functionality are the core framework layers' components. The social networking elements form the application component, and the goal is a modular design for extensibility. Furthermore, we conduct an in-depth study on P2P component mechanisms that fulfill the technical requirements. We then study several proposed P2P OSNs to compare the implementations and what they achieve to satisfy the user, system, and technical requirements.

We build our research on LibreSocial, a P2P framework for OSNs, that has been in continuous development since 2008. First we give a detailed description of LibreSocial, as it's structure has not yet been presented in that depth. We view LibreSocial as an ideal candidate for a P2P OSN, as it is easily extendable to address the challenging demands that we face. Additionally, LibreSocial fulfills the defined technical aspects using a modular design to achieve a zero-trust network. We then conduct detailed benchmarking tests on LibreSocial. The tests aim to reveal the interaction between the social network elements with one another and the underlying services that the P2P components provide. We also seek to discover LibreSocial's ability to operate in the wild, how well it scales up and handles churn while remaining stable, and how the storage characteristics, particularly the number of replicas (replication factor), affect the performance of the application. We show that social network elements synergize well with one another, and the underlying services offered by the P2P components provide adequate

support for the network. We also show that the network scales up to 2000 nodes without service degradation and handles churn well while remaining stable. Additionally, our results on the impact of the replication factor indicate a need to increase the number of replicas as the network scales up to maintain service quality.

As a way of improving service delivery, we propose three mechanisms. The first mechanism is metadata-based search techniques to solve retrieving of documents stored in a distributed data structure, such as a linked-list or set, using a multi-attribute query. We present two search techniques exhaustive and first-match, and four join algorithms Simple LocalJoin, Parallel LocalJoin, asynchronous NetworkJoin, and BloomJoin. We also consider three distributed data structures, binary tree, deep tree, and customized broad tree. We show that the appropriate combination of the search technique, join algorithm, and distributed data structure is necessary to achieve optimum local and network performance.

The second mechanism we propose is a social caching mechanism that takes advantage of social data to improve data availability. Using this mechanism, we aim to decrease the number of overlay requests by using the social interaction data to implement active dissemination for frequently accessed data and caching this in a social cache. We implement three strategies for selecting users whose data is to be cached, namely, random, trend, and social score selection strategies. We show that the social score strategy is the most advantageous. Using social score strategy, we implement a social cache mechanism and show that coupled with the regular cache, we can achieve 99% cache-based retrievals, but at some cost to the network due to the combination of active data dissemination and regular updates to the regular cache for freshness.

The last mechanism we propose is a capacity management protocol to address the social network's heterogeneous nature by differentiating strong nodes and weak nodes. In a social network, users connect with different devices having varying capacities (memory, bandwidth, and processing power). Further, the type of connection (LAN/WLAN or metered) is significant. Therefore low capacity devices or devices connecting via metered connections are considered weak nodes. To ensure a stable network, we aim at preventing the weak nodes from participating in routing and storing of replication data. We show that, with the capacity management protocol, this is achievable. We also show that it is possible to have up to 75% of the network composed of weak nodes and maintain a stable network, although there will be some service delivery degradation.

In conclusion, in this work, we identify and elaborate on the main concerns raised with the current popular OSNs leading to proposals for considering decentralized solutions, specifically P2P. We give the reasons for selecting the P2P platform and discuss the technical challenges of such a move. We then conduct a study to discover the necessary user and system requirements needed to define the technical requirements for a P2P framework for OSNs. We present LibreSocial as a P2P-based OSN that fulfills the specified technical requirements, and provide a systematic and large-scale evaluation of LibreSocial. We then propose three mechanisms for improving service delivery, metadata-based search techniques for distributed data structures, a social caching mechanism for enhanced data availability and faster retrievals, and a capacity management protocol to support heterogeneous nodes. With these contributions, we address the open questions raised in this Dissertation and enable LibreSocial to be utilized in use cases found in the real world.

Zusammenfassung

Die Nutzung der sozialen Medien im einundzwanzigsten Jahrhundert beeinflusst viele Bereiche des täglichen Lebens. Je nach den Vorzügen der Nutzer stehen verschiedene soziale Medienplattformen zur Verfügung, wie z.B. soziale Netzwerke wie Facebook und LinkedIn, Blogs z.B. Tumblr und WordPress, Mikroblogs z.B. Twitter und Videoplattformen wie YouTube. In den letzten zehn Jahren haben soziale Netzwerke viel Aufmerksamkeit erhalten, da sie auf eine zentralisierte Computerplattform angewiesen sind. Wir fassen die Bedenken in zwei Kategorien zusammen: kumulierte Kosten aufgrund der Zentralisierung sowie Sicherheits- und Datenschutzbedenken, die dadurch entstehen, dass ein einziger Anbieter alle hochgeladenen Daten kontrolliert und besitzt. Die Notwendigkeit, die Kosten auszugleichen, führt zu einer Monetarisierung der Benutzerdaten, was zu Bedenken hinsichtlich des Datenschutzes führt. Ein weit verbreiteter Vorschlag ist der Übergang zu einem dezentralisierten Modell der Datenverarbeitung, von dem wir erwarten, dass es die Probleme lösen wird. Ein föderiertes Netzwerk ist ein Bruch mit dem zentralisierten Modell, bei dem die Dienste von mehreren unabhängigen Anbietern bereitgestellt werden. Diese Dezentralisierung ermöglicht es dem Netzwerk auch widerstandsfähig gegen Zensur zu werden und reduziert die durch die Zentralisierung entstehenden Kosten deutlich, wenn sie nicht sogar vollständig.

Wir verfolgen die Idee der vollständigen Dezentralisierung der Systemarchitektur durch Einsatz des Peer-to-Peer (P2P)-Modells, um beiden Anliegen gerecht zu werden. Es überträgt die Infrastrukturkosten von den Anbietern auf die Nutzer, die nun direkt die Infrastruktur bereitstellen, um das Netzwerk am Laufen zu halten. Es stellt auch sicher, dass die Daten der Benutzer privat sind und zugriffsgeschützt auf den Geräten der Benutzer gespeichert werden. Außerdem stellt es sicher, dass das Netzwerk der Zensur standhält. Die Verwendung der P2P-Technologie bringt jedoch die Herausforderung mit sich, Mechanismen zu entwickeln, welche die Dienstverfügbarkeit so verbessern, dass sie der von zentralisierten sozialen Online-Netzwerken (OSNs) angebotenen entspricht.

Um die P2P-Technologie für die Implementierung von OSNs zu motivieren, untersuchen wir soziale Netzwerke im Allgemeinen, um die Benutzeranforderungen und funktionale und nichtfunktionale Systemanforderungen zu ermitteln. Aus diesen Anforderungen leiten wir die technischen Anforderungen in eine Vier-Komponenten-Architektur ab, die sich aus dem Overlay, dem Basis-Framework, den sozialen Netzwerkelementen und der grafischen Benutzeroberfläche zusammensetzt. Das Overlay bietet Unterstützung für Adressverwaltung, Routing und Sicherheit. Speicherung, Kommunikation, Suchmechanismen, Zugriffskontrollmechanismen und Überwachungsfunktionen sind die Komponenten des Basis-Frameworks. Die Elemente des sozialen Netzwerks bilden die Anwendungskomponente, und verfolgen das Ziel einer modularen Erweiterbarkeit. Für eine Übersicht, führen wir eine eingehende Studie über verschiedene Mechanismen der P2P-Komponenten durch, welche die technischen Anforderungen erfüllen. Dann untersuchen wir mehrere vorgeschlagene P2P-OSNs, die untersucht wurden, um die Implementierungen zu vergleichen und zu vergleichen, was sie leisten, um die Benutzer-, System- und technischen Anforderungen zu erfüllen.

Wir bauen unsere Forschung auf LibreSocial auf, einem P2P-Framework für OSNs, der seit 2008 kontinuierlich weiterentwickelt wird. Zuerst geben wir eine ausführliche Beschreibung an, da seine Struktur noch nicht in dieser Tiefe vorgestellt worden ist. Wir betrachten LibreSocial als idealen Kandidaten für ein P2P-OSN, da es sich leicht erweitern lässt, um die herausfordernden Anforderungenzu erfüllen. Daneben erfüllt LibreSocial die definierten technischen Aspekte unter Verwendung eines modularen Designs, um ein Zero-Trust-Netzwerk zu erreichen. An-

schließend führen wir detaillierte Benchmarking-Tests zu LibreSocial durch. Die Tests zielen darauf ab, die Interaktion zwischen den sozialen Netzwerkelementen untereinander und den zugrundeliegenden Diensten, welche die P2P-Komponenten bereitstellen, aufzuzeigen. Wir wollen als Ziel herausfinden, ob LibreSocial in der echten Welt funktionieren, wie gut es sich skalieren lässt, wie gut es Churn bewältigt und dabei stabil bleibt und wie die Speichereigenschaften, insbesondere die Anzahl der Replikate (Replikationsfaktor), die Leistung der Anwendung beeinflussen. Wir zeigen, dass die Elemente des sozialen Netzwerks gut miteinander harmonieren und dass die zugrunde liegenden Dienste, welche die P2P-Komponenten bieten, das Netzwerk angemessen unterstützen. Wir zeigen auch, dass das Netzwerk bis zu 2000 Knoten ohne Dienstdegradation skaliert und Churn gut verkraftet, da es stabil bleibt. Darüber hinaus weisen unsere Ergebnisse zur Auswirkung des Replikationsfaktors auf die Notwendigkeit hin, die Anzahl der Replikate zu erhöhen, wenn das Netzwerk skaliert, um die Dienstqualität aufrechtzuerhalten.

Als eine Möglichkeit zur Verbesserung der Dienstbereitstellung schlagen wir drei Mechanismen vor. Die ersten Mechanismen, auf Metadaten basierende Suchtechniken, ermöglichen das Auffinden von Dokumenten, die in einer verteilten Datenstruktur, wie z.B. einer verknüpften Liste oder Menge, gespeichert sind, unter Verwendung einer Multi-Attribut-Abfrage, indem sie die Suchtechniken und die Join-Algorithmen berücksichtigen. Wir stellen die zwei Suchtechniken Exhaustive und First-Match sowie vier Join-Algorithmen vor: Simple LocalJoin, Parallel LocalJoin, asynchroner NetworkJoin und BloomJoin. Wir betrachten auch drei verteilte Datenstrukturen, Binary Tree, Deep Tree und Customized Broad Tree. Wir zeigen, dass die passende Kombination aus Suchtechnik, Join-Algorithmus und verteilter Datenstruktur notwendig ist, um eine optimale lokale und Netzwerk-Performance zu erreichen.

Der zweite Mechanismus, den wir vorschlagen, ist ein sozialer Caching-Mechanismus, der sich soziale Daten zunutze macht, um die Datenverfügbarkeit zu verbessern. Mit diesem Mechanismus wollen wir die Anzahl der Overlay-Anfragen verringern, indem wir die Daten zur sozialen Interaktion nutzen, um eine aktive Verbreitung für häufig abgerufene Daten zu implementieren und diese in einem sozialen Cache zwischenzuspeichern. Wir implementieren drei Strategien zur Auswahl der Benutzer, deren Daten zwischengespeichert werden sollen: Zufalls-, Trend- und Social-Score-Auswahlstrategien. Wir zeigen, dass die Social-Score-Funktion die beste Lösung aus den dreien bietet. Unter Nutzung der Social-Score-Strategie implementieren wir einen Social-Cache-Mechanismus und zeigen, dass wir in Verbindung mit dem regulären Cache 99% cachebasierte Abrufe erreichen können, allerdings mit gewissen Kosten für das Netzwerk aufgrund der Kombination aus aktiver Datenverbreitung und regelmäßigen Aktualisierungen des regulären Caches.

Der letzte Mechanismus, den wir vorschlagen, ist ein Kapazitätsmanagementprotokoll, um die heterogene Natur des sozialen Netzwerks durch die Unterscheidung zwischen starken und schwachen Knoten zu behandeln. In einem sozialen Netzwerk verbinden sich die Benutzer mit verschiedenen Geräten mit unterschiedlichen Kapazitäten (Speicher, Bandbreite und Verarbeitungsleistung). Darüber hinaus ist die Art der Verbindung (LAN/WLAN oder Volumentarif) von Bedeutung. Daher gelten Geräte mit geringer Kapazität oder Geräte, die über volumenbasierte Verbindungen verbunden sind, als schwache Knoten. Um ein stabiles Netzwerk zu gewährleisten, wollen wir verhindern, dass sich die schwachen Knoten an der Routing und Speicherung von Replikationsdaten teilnehmen. Wir zeigen, dass dies mit unserem Kapazitätsmanagementprotokoll erreichbar ist. Wir zeigen auch, dass es möglich ist, bis zu 75% des Netzwerks aus schwachen Knoten zusammenzusetzen und ein stabiles Netzwerk aufrechtzuerhalten, auch wenn es zu einer gewissen Verschlechterung der Dienstbereitstellung kommt.

Abschließend zusammengefasst stellen wir in dieser Arbeit die Bedenken dar, die gegenüber den

derzeit populären OSNs geäußert wurden und zu Vorschlägen für die Erwägung dezentralisierter Lösungen, insbesondere P2P, führen. Wir geben die Gründe für die Wahl der P2P-Plattform an und diskutieren die technischen Herausforderungen eines solchen Schrittes. Unser Survey zu dezentralen OSN ermittelt die Benutzer- und Systemanforderungen, die erforderlich sind, um die technischen Voraussetzungen für ein P2P-Framework für OSNs zu definieren. Wir stellen LibreSocial als ein P2P-basiertes OSN vor, das die spezifizierten technischen Anforderungen erfüllt, und durch eine Reihe von Benchmarking-Tests zeigen wir, dass LibreSocial in realistischen Szenarien mit bis zu 2000 Nutzern stabil, performant und kostengünstig funktioniert. Wir schlagen außerdem drei Mechanismen zur Verbesserung der Dienstbereitstellung vor, metadatenbasierte Suchtechniken für verteilte Datenstrukturen, einen Social-Caching-Mechanismus für eine verbesserte Datenverfügbarkeit und schnellere Abrufe sowie ein Kapazitätsverwaltungsprotokoll zur Unterstützung heterogener Knoten. Mit diesen Beiträgen gehen wir auf die offenen Fragen ein, die in dieser Dissertation aufgestellt wurden, und ermöglichen die Nutzung von LibreSocial in Anwendungsfällen, die in der realen Welt anzutreffen sind.

Acknowledgements

I would like to take this opportunity to thank the people who have been a great source of encouragement, inspiration and support during my time as a doctoral student. To my supervisor Kalman Graffi, I am grateful for the chance to undertake my research work under your tutelage, an experience that has left a life-long impression, and for your enthusiasm and encouragement when there seemed to be no light at the end of the tunnel. I would also like to thank my mentor Stefan Conrad for always giving a listening ear and supporting me in the time of need.

I am also grateful to the entire team of the Computer Networks department and, in particular, to the members of "Technik sozialer Netzwerke" work group for a friendly working environment. To Sabine Freese, it was a pleasure working with you, especially in navigating the institutional system. Special thanks, in no particular order, to Andreas Disterhöft, Andre Ippisch, Raphael Bialon, Ahmad Cheraghi, Raed Al-Aaridhi, Alexander Schneider, and Christian Meter for the interesting intellectual discussions over lunch and in other interactions. To my first office mate, Tobias Amft, thanks for welcoming me. To my current office mate, Ahmad Rabay'a, it was a pleasure to interact with you at various levels, both academic and general.

I would also like to thank all the committed students who supported me in my research during their final theses or research projects: Iakov Dlikman, Philipp Mainz, Moritz Kanzler, Liat Khitman, Sebastian Bischoff, and Moritz Gericke.

I am most grateful to my parents and siblings for their support and checking up on me. This journey would not have been easy if it were not for my wife, Pauline. Your love, encouragement, and support in the hardest of times is immeasurable.

Last, and most importantly, I am grateful to God Almighty for life, for every opportunity and for every step made this far.

Contents

Chapter 1

Introduction

Social media has enriched the number of options available for communication, presenting new opportunities for users to connect and communicate with old friends and find new ones. It has been a major factor in influencing several changes in today's cultural and moral perceptions. By leveraging social media, old friends previously separated by time and space have reconnected, opinions on sensitive societal issues under discussion are shared and sometimes altered, and businesses have succeeded in increasing their visibility, increasing their market share. However, conceptualizing what exactly social media is may not be immediately feasible for two reasons [116]. The first is that the rates of technological expansion and evolution present a challenge in giving a clear definition. The second is that social media services facilitate communication forms offered by other technologies. Notwithstanding, there is a general agreement on the distinguishing characteristics of social media: social media services are Web 2.0 Internet-based applications; user-generated content forms the core of social media; the need for creating user-specific profiles that are maintained by the service provider and the services encourage the formation of an online social network (OSN) by connecting the user-profiles of different individuals/groups [116]. Thus, a working definition of social media is interactive computer-mediated technologies that allow users to freely express themselves by creating/sharing information, ideas, or career interests via virtual communities and networks [81]. Without the development of Web 2.0, all this would not have been possible.

Web 2.0, the Social Web and Online Social Networks

The proposal for the development of the *World Wide Web (WWW)*, (or Web), by Tim Berners-Lee [13] was a significant step in online communications and information diffusion. The Web's impetus was to provide a standard format for documents stored on servers and provide each document with a unique name to locate and retrieve the document via a Web browser program [110]. The term Web 2.0 was first used by Darcy DiNucci [32] and then by Tim O'Reilly in 2004 [114, 115]. Web 2.0 is *"a collection of open-source, interactive and user-controlled online applications expanding the experiences, knowledge, and market power of the users as participants in business and social processes. Web 2.0 applications support the creation of informal users' networks facilitating the flow of ideas and knowledge by allowing the efficient generation, dissemination, sharing and editing/refining of informational content"* [26]. Web 2.0 made it possible to introduce other services on the Internet, such as blogs, wikis, multimedia sharing services, content syndication, podcasting, and content tagging services [7]. These services have made it possible to launch other technologies such as social networking, aggregation services, data 'mash-ups', tracking and filtering content, collaborating, replicate office-style software in the browser, and source ideas or work from the crowd [7].

Term	Source	Definition
Computer-supported social networks	[149]	"When computer networks link people as well as machines, they become social networks."
Mediated publics	[16]	"Environments where people can gather publicly through mediating technology."
Online social network	[60]	"An online platform that (1) provides services for a user to build a public profile and to explicitly declare the connection between his or her profile with those of the other users; (2) enables a user to share information and content with the chosen users or public."
	[25]	"An online service or site to facilitate social interaction to help individuals find others of a common interest, establish a forum for discussion, and exchange information."
	[64]	"Results from the use of a dedicated web-service, often referred to as social network site (SNS), that allows its users to (i) create a profile page and publish messages, and (ii) explicitly connect to other users thus creating social relationships. *De facto*, an OSN can be described as a user-generated content system that permits its users to communicate and share information."
Social networking services	[3]	"Social networking services gather information on users' social contacts, construct a large interconnected social network, and reveal to users how they are connected to others in the network."
Social networking sites	[17]	"Web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system."

Table 1.1: OSN definitions

Defining the Social Web and Online Social Networks

Even though previously, the Internet and the Web always facilitated social interaction, Web 2.0's emergence and speedy diffusion of its functionalities were an evolutionary step forward in the social aspect of Web usage [116]. As a result, there is a rise in the popularity and consequent social media usage, resulting in the *Social Web phenomenon*, the Web of Social Media [111]. The Social Web is *"the platforms, technologies, and applications that enable the Web to support and foster social interaction"* [111]. In discussions on social media usage, it is impossible to ignore the impact and contribution of OSNs, and social networking in general, since they form the bulk of social media usage. Social networking is the act of engagement in which people with common interests associate together to build relationships through community [68]. Taking into account the Internet as the interaction platform, then social networking is the creation and maintenance of personal and business relationships with emphasis on online communications [134]. However, a single well-established definition for OSNs may not be directly possible, a characteristic of Web 2.0 applications [69]. Several other terms also refer to them, such as mediated publics [16], computer-supported social networks (CSSN) [149], social networking sites [16, 17], social networking services [3], web-based social networks [49], and virtual communities [23]. Table 1.1 presents some of the common definitions in the literature. For OSN providers, the aim is to gain a stable user community. Therefore, the application developers must understand what the users want. In the next section, we discuss the user requirements to consider when developing an OSN.

OSN Requirements

For users of OSNs, the implementation of and interactions between the system's various components are not of concern. The users' interests are on the benefits that accrue from accessing the application. Therefore, to provide the services, the developers of the OSNs must carefully consider the functionalities and the privacy and security features that the system should cover.

OSN functionalities

The purpose of using OSNs can be summed up into two aspects, keeping in touch with other users and identity management [126]. OSN interactions are different from regular social interactions between people in the real world based on the fact that in OSNs (i) no physical contact between communicating parties occurs, (ii) there is a lack of unambiguous and reliable correlation between the Internet and real-world identities, (iii) multi-modal communication with several parties is possible and it is easy to switch between the different communication channels, (iv) it is easy to break-up and suspend contacts or relationships, (v) present an opportunity for gathering and mining the data about communications and other activities, and (vi) have low reliability about the information users provide about themselves due to privacy concerns [107]. For this to be possible, [156] points out that the OSN should support the construction of a digital representation of users (user profiles), allow the users to define their social connections via contact lists, support maintenance and improvement of the social connections among users given the physical and virtual world, and provide a means for finding new connections based on criteria such as interests, locations, and common friends. Therefore, we can see the need for identifying the core functionalities OSNs should incorporate. [4, 126, 156] further elaborates on these functionalities, summarized in the following.

a) *Personal space management* - The OSN should allow users to register or withdraw from the platform, create and edit user profiles, and upload and share the user-generated content (UGC) such as photo images, audio/video clips, and blog posts.

b) *Social connection management* - The OSN should give users the ability to establish, maintain, and revoke social connections. The users should also define the type of relationship they have with other users, for example, using friend lists. In essence, it establishes the level of "connectedness" between users.

c) *Means of communication* - The ability to exchange information directly or indirectly between the users is the backbone of OSNs. It should incorporate synchronous communication, such as instant messaging, and asynchronous communication, such as private messaging.

d) *Shared storage space interaction* - The OSN should allow users to interact with one another via a shared space such as via walls, view their contacts' wall updates, know the current status and changes to the contacts' status, and interact in forums and share documents.

e) *Social graph traversal* - The users should be able to traverse the online social graph, for example, by examining other users' friend lists. Using a traversal policy will restrict who is allowed to explore a user's friend list.

f) *Search facilities* - The users should have the ability to explore the social network space to find and connect to other users. Two types of search techniques are distinguished:

searching based on criteria such as name, gender, interests, and location; and pro-active reception of recommendation for interesting contacts by the OSN [126].

Even if these functions are present in the OSN, users' privacy and security must be guaranteed by coupling these functionalities with appropriate security and privacy mechanisms. We, therefore, present the privacy and security considerations for OSNs.

Privacy and security in OSNs

The standard security considerations for a network usually include confidentiality, integrity, and availability. However, for OSNs, the combination of usability and sociability, and privacy and security presents a challenge that requires a delicate balance [156]. Following is a discussion on the core privacy and security requirements.

Privacy requirements This addresses illegal disclosure and inappropriate use of private information without authorization [156]. The main privacy risks emanate from users divulging personal data, inflexible privacy tools in the OSNs, and lack of sufficient access control mechanisms to mitigate unauthorized access to their profile information [70]. Thus the goal of privacy is to protect the user's identity (identity anonymity) and ensure no possibility of linking several private data files to the profile of the owner (unlinkability) [156]. By incorporating mechanisms to guarantee confidentiality, ownership privacy, social interaction privacy, and activity privacy [4], the OSN fulfills the privacy requirements. Typically, users would like to define with whom they share their data, and sometimes the infrastructure/platform/ operator requires them to open the list of recipients further.

Security requirements As OSNs are network applications, they are open to security risks. These security risks can either be attacks on the users or the OSN [80]. Security risks in general focus on two aspects, that is the data and the communication channel. For this reason, the OSN must account for the security requirements by the inclusion of mechanisms that ensure channel availability, channel authentication, data integrity and authenticity, and non-repudiation [4].

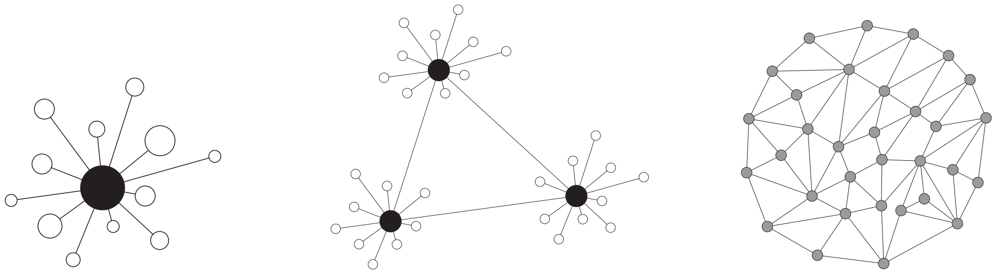
With the core OSN functionalities and privacy and security considerations addressed, the next concern is the architectural models used for the OSNs.

OSN architectural models

In the design of the OSNs, the components that present the most significant concern are content storage, access control, and interaction and signaling. These components' design depends on the architectural models, broadly classified into two groups, centralized and decentralized models [118]. We consider each of the two classifications separately.

Centralized models

The main distinguishing feature of the OSNs based on this model is the central service provider, hence single administrative control, shown in Figure 1.1a. This model is the most commonly used for OSN applications. With this model, the content storage, access control, and interaction and signaling mechanisms are in the service provider's hands. The upside to this model is



(a) Centralized (b) Federated (c) Peer-to-peer

Figure 1.1: OSN architectural models

that it gives the developers complete autonomy. Therefore, the OSN design process is simple. It also guarantees service availability because of dedicated resources and makes it possible to have quick system/software updates. Thus, OSNs developed on this architecture offer high service quality, and with an application programming interface (API), the inclusion of third party applications for more functionalities is possible. However, on the downside, scaling and maintaining the system is an expensive task, often imposing a substantial financial burden on the OSN providers, leading to monetization of users' data that the provider wholly owns. This action presents a serious privacy risk to the users by opening them to targeted advertising because of personal data markets [137], possible emotional manipulation [9, 88], and online activity surveillance [10, 132, 140]. The centralized model is also a strong candidate for censorship, for example, in times of political instability, such as during the Arab Spring [75]. Despite the advantage of high service quality gained, the infringement on the users' privacy presents a serious challenge leading to considerations for alternative models for developing OSNs.

Decentralized models

The OSNs in this classification have multiple mediating nodes. The decentralization generally focuses on three aspects: storage of content, access control mechanisms, and interaction and signaling mechanisms [120]. The decentralized models can be either federated or peer-to-peer (P2P) models.

Federated model In this model, the provision of services is decentralized across several distinct providers, as shown in Figure 1.1b, to form a federated network. A federated network consists of multiple social network sites implementing open standard protocols that allow them to interact. Usually, each social network comprises several independently owned server nodes connected to the federated social network. Users then connect to the social network and server node of their choice. An example of a federated network is Fediverse[1] (a portmanteau of federated and universe), including networks like Diaspora, Mastodon, Pixelfed, Hubzilla, GNU Social, and PeerTube, among others. The storage of content is partially decentralized, as the independently owned servers still store the users' data and private information. To address access control, most offer fine-grained access policy mechanisms that allow users to control the visibility of what they post, and some such as OneSocialWeb enable access control for profiles and relationships. The mediating servers handle the interaction and signaling mechanisms in

[1] https://fediverse.party/

the federated OSNs, meaning they are not fully decentralized. OSNs that are designed on the federated model present some advantages. There is reduced management of social relationships and a significant reduction in the costs associated with managing the infrastructure. The network also encourages free speech. The disadvantage with this system is that the server nodes present a privacy risk as they store the user's private information, and they may also monitor the interactions between the users. Because the server nodes' maintenance and administration still present a financial burden, the monetization incentive still exists but at a much lower degree than in the centralized model. Also, the quality of service of the application is highly dependent on the end-users' resources. Thus, although this model's use reduces the financial burden due to infrastructure and maintenance costs and has the potential to be more scalable than the centralized OSNs, it does not altogether eliminate the financial implications and still does not address the privacy concern.

Peer-to-peer model In this model, every node simultaneously acts as a server and a client, hence a fully decentralized network as in Figure 1.1c. The users provide all the resources required to support the OSN, eliminating the financial burden due to a single provider (centralized model) or multiple independent providers (federated model). The application utilizes existing P2P technologies such as the FreePastry library[2] based on Pastry [129], or by writing new P2P protocols. Usually, only an installation of the application is needed, and no additional technical settings are involved. The users fully control what is stored on the system, such as a distributed hash table, with redundancy mechanisms applied for guaranteed data resilience and encryption for data security. Therefore, user privacy management is also decentralized into the users' hands, and cryptography plays a crucial role in the system, not the users' goodwill. The P2P model is also resilient to censorship and guarantees freedom of interactions. Thus, in P2P OSNs, storage of content, access control mechanisms, and interaction and signaling mechanisms are fully decentralized. However, the P2P-based OSNs still exhibit challenges that developers must critically address. They will then have the ability to offer users quality of service comparable to centralized OSNs in system performance, the privacy of content and interactions, the richness of functionality, and be economically viable [120]. P2P OSNs exhibit two bootstrap problems [78]. The first is the need for enough bootstrap nodes that are openly reachable through public IP addresses. The second is the problem of peer discovery and social relationship management. Other challenges include guaranteeing profile and content availability due to intermittent connectivity; efficient content distribution; efficient resource and energy utilization, especially when mobile devices are present; ensuring users' security and privacy for both communication and data; and proof of ability to scale [94]. We view the P2P model as a legitimate solution due to the decentralization of infrastructure provision, which eliminates the financial burden. By allowing users to store their data and control their privacy, it also addresses privacy concerns. However, the challenges in the P2P OSNs require novel solutions that lead to desired performance.

1.1 Motivation

At a minimum, an OSN should include the core functional requirements and incorporate appropriate privacy and security measures. A good blend of these requirements leads to a high quality of service and increases users' trust in the system. Centralized OSNs offer high-quality

[2] https://www.freepastry.org/FreePastry/

services and present a wide variety of functionalities to users, who implicitly must trust that the providers will not misuse the data they upload. However, to meet high operational costs incurred due to the system scaling and address the need to realize profit margins, centralized OSN providers monetize users' data that they fully own, breaching the users' trust. On the other hand, federated networks present a change from a single administrative domain to multiple independent mediators. Federated OSNs are a workable solution to expensive scaling due to rising infrastructural and system maintenance costs. However, they do not solve the problem of user data privacy. Furthermore, the costs associated with keeping the server nodes running still exists, leading to possible user data monetization. Thus, to mitigate the two compounded problems, that is, eliminate the operational costs and address the privacy concerns, any proposed solution needs to be: *a*) be financially sustainable, *b*) solve the privacy and security challenges, and *c*) support seamless dynamic growth.

Therefore, the third architectural model, peer-to-peer, is aimed at, as it shifts infrastructure provision to the users and allows them to store their data and control their privacy. However, the design of a private and secure P2P OSN presents several challenges that can be grouped into three categories, as highlighted by [120].

Decentralized content storage This problem also directly impacts the data availability, cost of storage, and trust. For P2P OSNs, data availability is strongly affected by the peers' online times storing the data. Mechanisms such as replication and caching help improve availability. The storage costs are influenced by the storage space that peers are willing to share with the network and the users' bandwidth resources. Trust focuses on whether the replicating node can trust the nodes receiving the replicas. One method to handle this is storing at friends' node. We envision a zero-trust P2P OSN where friends are also not trusted to handle data correctly and therefore, any storage options must include appropriate encryption schemes.

Decentralized access control This presents the challenge of limiting access to content to only designated recipients, therefore enforcing privacy. Access control solutions are access control lists (ACLs), encryption algorithms, or both. When using ACLs for access restriction, there is a need for an authentication mechanism coupled with an appropriate asymmetric encryption algorithm. For decentralized OSNs, to ensure encryption key distribution remains fully decentralized, the goal is to avoid using Certification Authorities (CAs) and public-key infrastructure (PKI). One possibility is to use a DHT substrate to distribute the keys [57].

Interaction and signaling mechanisms P2P systems usually rely on the overlay to handle signaling and mediate interactions. In distributed hash table (DHT) based OSNs, the DHT manages asynchronous messaging and allows users to perform exact match query on the DHT using an ID. P2P-based OSNs also allow for direct interaction between peers, even in the absence of an Internet connection. The security setup should include appropriate encryption algorithms to guarantee zero-trust communications.

To address these challenges, we thus present the problem statements and our research questions.

1.2 Problem Statement and Research Questions

The conceptualization and eventual realization of a working P2P-based OSN will entail handling four challenges, which we elaborate as problem statements PS1 to PS4.

PS1: Requirement analysis for the design of a P2P framework for OSN. As OSNs usage has grown, many of the features they offer have become standard. These features include personal and shared storage space, social connection capabilities to establish friendships or follower-ship, a suitable and secure communication channel, and search facilities. Besides, in an increasingly security-aware user base, the security and privacy features need to be incorporated at all application levels. To realize these features while meeting or surpassing centralized OSNs in terms of quality of service, developers need clarity on the expected OSN requirements from the beginning. Therefore, this research's first concern is to study the literature available to identify the requirements for a functional OSNs. For this, the services offered in centralized OSNs form a good starting point. These requirements can then be adapted for a decentralized solution, in particular, for a P2P-based solution. There may be a need for considering certain compromises to meet the identified requirements within the context of a decentralized system. From this, the following questions arise:

RQ1.1 *Based on the usage observed in the popular OSNs and presented in the literature, which OSN features are necessary from a user's perspective?* This question is fundamental in establishing a basis for orienting the solution to meet the users' identified needs.

RQ1.2 *What are the system design requirements (functional and non-functional) necessary to ascertain a working OSN?* This question builds up from the OSN features, and by translating them into system functions, by intuition, the designers can project the non-functional requirements.

PS2: Description of a software architecture for a P2P framework for OSNs. With the core system requirements for a secure and reliable OSN defined, the next step is to identify the building blocks for hosting the OSN in a P2P platform, which will help us choose the component mechanisms for inclusion in the application. By decentralizing the functionalities of the OSNs, the system designers must consider mechanisms that tackle network aspects such as robustness under churn, handling distributed data storage management and update propagation management, define the overlay topology and a protocol for searching and addressing among other challenges [63]. The following are the research questions that arise.

RQ2.1 *What are the principal technical requirements that guide the modeling and eventual development of the OSNs software architecture?* There is a need to translate the system design requirements into technical requirements to explicitly define the P2P components for incorporation in the software architecture.

RQ2.2 *What are the alternative P2P component mechanisms available to meet the need expressed by each defined technical requirement?* Based on the alternative mechanisms, what would be a good fit when developing the OSNs? These questions guide us in identifying the different component mechanisms available in the literature to implement each functionality.

RQ2.3 *How do alternative P2P-based OSNs implement the OSN functionalities they offer, and how do these OSNs compare against the defined technical requirements?* What compromises, in terms of the OSNs features and system requirements, do these implementations make? How workable are they in the real environment? Here, we hope to identify several proposed solutions, analyze their makeup, and compare them based on the services that they offer.

RQ2.4 *To what extent can we translate the proposed software architecture into a working OSN* *that meets the defined technical requirements?* The desire is to show the viability of converting the technical requirements and, therefore, software architecture into a working solution.

PS3: Decentralized mechanisms for reliable, scalable, and secure OSNs. With the proposed OSN in place, the next challenge is introducing mechanisms that improve the application's service delivery in a decentralized environment. Such improvements may include efficient data retrieval by reducing retrieval time or reduced network traffic by improving the mechanism used for data retrieval. The improvements will help ensure that P2P OSNs service quality compares well against centralized OSNs, while still guaranteeing security and privacy. To show that the application is extendable, we aim at a modular design. We consider the following research questions.

RQ3.1 *How can we utilize the metadata to improve search and retrieval of documents stored in complex data storage mechanisms such as distributed data structures (DDSs) in the network?* Which techniques can improve retrieval time? How does the type of DDS impact the retrieval process?

RQ3.2 *To what extent is it possible to improve data availability by leveraging social data?* What are the compromises to contend with, considering the overall performance?

RQ3.3 *With users connecting to the network using heterogeneous devices, to what extent can the network distinguish these nodes based on their capacities to ensure a stable network?* How will low capacity devices be managed in the overall network management scheme considering routing and replication?

PS4: Systematic evaluation of interdependencies in the P2P framework for OSNs With all vital building blocks now incorporated, the next consideration is to perform individual evaluations of the mechanisms, such as estimating the quality of a specific distributed data storage approach, and combined evaluations of all mechanisms. These evaluations aim to systematically investigate the interdependencies of the various system components related to overall performance. We seek to answer the following research questions.

RQ4.1 *How do the actions initiated by the various social network elements impact the system's performance?* This question focuses on the interaction between the social network elements and the underlying layers.

RQ4.2 *How well does the system adapt itself in a realistic environment?* This question seeks to find out the viability of the system in a real-world context for eventual deployment.

RQ4.3 *How scalable and stable is the proposed P2P framework for OSNs?* How does it behave under adverse network dynamics such as high churn? By this, we seek to establish the scaling bounds of the system. By extension, we also strive to show a stable system during scaling up/down.

RQ4.4 *How does the storage dynamics affect the system's performance in general?* As one of the challenges is to ensure that the user's data is always available, we seek to see how the system behaves when the number of replicas for a given item is varied.

Having presented the four problem statements with the associated research questions handled in this thesis, in the next section, we embark on showing the contributions made towards tackling the research questions.

1.3 Contributions

We document the achievements realized from this thesis in six (6) publications that provide answers to the research questions presented in Section 1.2. These publications form the body of the succeeding six chapters (Chapters 2 to 7). In the following, we summarize how these publications address the research questions.

PS1: Requirement analysis for the design of a P2P framework for OSN - [98] To understand the need for decentralized online social networks (DOSNs), we first summarize the shortcomings of the popular centralized OSNs. In [98], we discuss the two main concerns raised against them, accumulated costs of centralization and security and privacy concerns, which motivate the move towards decentralized OSNs. We identify P2P OSNs as a viable solution because they eliminate infrastructure and maintenance costs and allow users to control their security and privacy. To tackle research question RQ1.1, [98] investigates related works to identify the desired OSNs features. These features are identity management, ability to converse, ability to share, provision for awareness of others' presence, relationship management, and ability to create groups. With these features now clear, we answer the research question RQ1.2 by converting them into system (functional and non-functional) requirements. The functional requirements are personal storage space management, social connection management, social graph traversal, communication channel, shared storage space interaction, and search facilities. The non-functional requirements ensure a secure system by offering privacy and security functions, and also, the inclusion of a metering functionality helps developers undertake system monitoring for performance improvement through a control loop.

PS2: Description of a software architecture for a P2P framework for OSNs - [98] and LibreSocial [55] With the system requirements outlined, we address research question RQ2.1 by defining a set of technical requirements that state the components for considerations. We then convert the technical requirements and summarize them into a four-fold software architectural model consisting of: the overlay, which defines node addressing, routing and undertakes the security functions; the storage and communication framework, which incorporates features for data redundancy (both simple data and complex data stored in DDSs), data access control, searching and retrieval, communication features such as direct, multicast and publish and subscribe, and the monitoring function; the social network elements implemented as plugins (mandatory and optional); and the graphical user interface (GUI). We discuss a suitable P2P component for each technical requirement, for which several alternative mechanisms are possible. By investigating and discussing these alternatives, we address the research question RQ2.2. The problem raised by research question RQ2.3 entails undertaking an analytical study of several proposed P2P-based OSN applications. Therefore, we give a detailed discussion of the identified proposals to show how they implement each technical requirement aspect. The application LibreSocial has been under iterative development from 2008, and since 2017, the author of this thesis has been leading the development of the framework. We present this as an answer to research question RQ2.4 and show that chosen the P2P components integrate into the software architecture proposed to answer research question RQ2.1 synergize well, attested by the experimental data analyzed.

PS3a: Metadata-based search algorithms for distributed data structures - [1] As a way to improve service delivery in the P2P OSN, we propose three mechanisms. The first mechanisms we present are metadata-based search algorithms for DDSs in [1] to answer research question RQ3.1. We utilize the metadata information to improve data search and retrieval in DDSs. We implement two searching techniques, exhaustive and first match, for data search and retrieval. We also implement four data join approaches, simple LocalJoin, parallel LocalJoin, asynchronous NetworkJoin, and BloomJoin, for consolidating the retrieved data to fulfill the query predicates and present the complete query. We test these on three DDSs, binary tree, deep tree, and customized broad tree. The results show that the choice of a suitable join algorithm depends on the type of DDS implemented and the search mechanism, and there is a need for careful considerations on compromises to make between performance and costs.

PS3b: Social caching mechanisms for improved data availability - [99] The second mechanism we present is a social caching algorithm in [99] as a response to the research question RQ3.2. The algorithm solves active data dissemination in DHT-based OSNs by utilizing the social data to select users whose data is stored in a social cache that supports active data dissemination. We propose and implement three selection strategies, random, trend, and social score. We select the social score selection strategy as the preferred choice due to better resource utilization and we use this to implement the social cache mechanism. We evaluate the social cache's effect on local and network performance by comparing performance without a cache, with the normal cache only, with the social cache only, and with both the social and normal cache working together. We note a cache hit ratio of 99% with the normal and social cache combined.

PS3c: Capacity management protocol for heterogeneity management - [97] The third service delivery improvement contribution we present is as a capacity management algorithm in [97] to answer the research question RQ3.3. We develop the algorithm to assist the OSN to monitor the different capacities of the user's device (node) such as bandwidth, type of connection (LAN/WLAN and metered), memory, and storage space available, processor capacity, and battery level. We then use this data to classify the nodes as weak or strong. The weak nodes are restricted from participating in routing and from storing replicated data. We implement our algorithm and test the network's ability to handle an increase in weak nodes. We show that the network remains stable even when the number of strong nodes in the network is as low as 1/4 of the total nodes, albeit with a reduction in the performance.

PS4: Systematic evaluation of interdependencies in the P2P framework for OSNs - [100] We address the last problem statement by performing a comprehensive evaluation of LibreSocial. This analysis is possible through the monitoring function integrated into LibreSocial called SkyEye [51, 54]. The questions raised under this problem are then individually addressed. We design a benchmarking model for P2P applications, which we use to test various system aspects. We conduct our tests in the high-performance cluster (HPC) computing environment provided by the "Centre for Information and Media Technology" (ZIM)[3] at Heinrich Heine University. We tackle the research question RQ4.1 by conducting experimental tests and analyzing the data generated by each action. The aim is to determine the maximum time to complete the action, the number of messages and data generated due to the actions, and the

[3] https://www.zim.hhu.de/high-performance-computing.html

message and data rates due to the actions. Further, we seek to show the interaction between the OSN elements (plugins) with the P2P mechanisms using this set of tests. To tackle research question RQ4.2, we develop a pseudo-random model for running the test based on network users' anticipated behavior over a given period based on the work in [11]. The pseudo-random model ensures that the OSN plugins are used in the same frequency and ratio as a regular usar would do. As a response to RQ4.3, we run an experiment that applies a step-wise incremental network join and thereafter, a step-wise network churn. In both cases, we perform loading after a stabilization period to ascertain network stability. The scaling up represents an increment of double, $^1/_2$, and $^1/_3$ of network's total number of nodes. The scaling down represents network churn of $^1/_4$, $^1/_3$, and $^1/_2$ of the network's total number of nodes. We show that the network remains stable in both the growth and churn phases with no recorded node failures. Finally, to address the research question RQ4.4, we focus on the impact of replicas per item in the network by varying the replication factor's value. We conduct experimental tests to show how storage performance is affected by changing the replication factor.

In the next and final section, we give an outline of the entire thesis.

1.4 Outline

This chapter has introduces and motivates decentralized solutions for OSNs and outlines the relevant research questions. The structure of the thesis is as follows:

Chapter 2 gives our contribution in terms of a comprehensive survey of P2P-based OSNs. We present the principal OSNs features, the system requirements, and the technical requirements for a P2P-based OSNs. Further, we discuss the P2P components needed to realize a working P2P application and their security implications and highlight the suitability for incorporation in an OSNs. We then compare several proposed P2P-based OSNs, paying attention to how they address privacy and security and the P2P components that are used to achieve a working solution.

In Chapter 3, we present LibreSocial, a P2P-based OSN framework that has been in iterative development since 2008 by various research groups and individuals under the direct supervision of Prof. Kalman Graffi, and whose development has been lead by the author of this thesis since 2017. We show that LibreSocial follows the proposed software architecture design based on the defined technical requirements. LibreSocial is a DHT based solution, relying on a heavily modified FreePastry overlay. On top of the overlay, a framework layer includes P2P mechanisms needed for the OSNs layer. We discuss how these layers interact and the different components in each layer leading to a working application. This chapter presents LibreSocial as it stood at the beginning of this work, the author of this thesis introduces any modifications, bug fixes, or improvements for better service delivery.

As proof of concept, we present a systematic evaluation of LibreSocial in Chapter 4. The aim is to show that the components of LibreSocial synergize well to ensure a working OSN with acceptable performance under loading conditions (messaging rates, storage rates, and retrieval rates) and acceptable cost implications (storage, memory, and bandwidth). We present experimental test results to show that the system is stable, scalable, and functions with minimal errors. The tests we conduct on LibreSocial are the first of their kind based on the experimental model (benchmarking) and the size as compared to other P2P OSNs

Chapter 5 presents metadata-based algorithms for data searching and retrieval in DDSs. We discuss two types of search techniques and four types of join algorithms for consolidating the retrieved documents to answer the final search query. We present these algorithms on the backdrop of three types of DDSs and show that depending on the DDS, the search technique and the join algorithms must be carefully selected.

In Chapter 6, we present a social caching mechanism that addresses the problem of efficient data management. We discuss three selection strategies, random, trend and social score, to employ in choosing users to subscribe to for active data dissemination, and show that the social score is advantageous. We also show that using the combined solution of the normal cache and the social cache mechanisms gives the best results but with some additional costs incurred in the network.

Chapter 7 presents a novel algorithm to handle capacity management in a DHT-based OSNs application such as LibreSocial by utilizing the device information to assign the device a network status. By analyzing the capacities of the devices, and the connection type, the nodes are classified as weak or strong. The weak nodes are barred from performing actively in routing and storing replication data, which are handled by the strong nodes.

Finally, we conclude our contributions and give our deductions on DOSN, focusing on P2P OSNs, and discuss possible future work in Chapter 8.

Chapter 2

Peer-to-Peer based Social Networks: A Comprehensive Survey

This chapter summarizes our contributions in [98] and gives a verbatim copy at the end.

Newton Masinde and Kalman Graffi. “Peer-to-Peer based Social Networks: A Comprehensive Survey”. In: *SN Computer Science* 1.5 (2020), pp. 1–51. DOI: 10.1007/s42979-020-00315-8

In this chapter, we summarize the contents of our paper [98], a study on the state-of-art on peer-to-peer based online social networks. The summary proceeds as follows. In Section 2.1 we present the paper summary. We then give this work’s contributions in Section 2.2, present the personal contributions in Section 2.3, and finally give the importance and impact on this thesis in Section 2.4.

2.1 Paper Summary

The Internet has grown in leaps and bounds since the turn of the twenty-first century, , fueled by two key stimuli: *technological advancements* and *user needs* [152]. The change from Web 1.0 to Web 2.0 [106, 114] facilitated the introduction of new and better services on the Web that were not possible before. A service that has grown significantly in its utilization is the online social network (OSN) services. For example, between 2008 and 2018, Facebook grew from 100 million users to 2.26 billion users (+2,155%), and YouTube grew from 1.95 million users to 1.9 billion users (+97,520%)[1]. In the case of Facebook, we see a linear growth of about 500 million new uses every 2.5 years over the same period. The online social networks (OSNs) by design mimic real-life social networks by providing a virtual platform for users to interact and connect. In what follows, we give a brief analysis of the concerns raised against the popular OSNs and the proposed solutions leading to our preferred choice, peer-to-peer (P2P) based OSNs. Afterward, we give a brief analysis of the appropriate technical requirements for a peer-to-peer (P2P) framework for OSNs. We follow this up with an in-depth study of the P2P components necessary for the proposed framework’s proper functioning. we present a summary of the P2P-based OSNs that we studied.

[1] https://ourworldindata.org/rise-of-social-media

2.1.1 Online Social Network

The first popular OSNs offered users limited services such as MySpace[2] for sharing music videos and audios, ICQ[3] and AOL Instant Messenger (AIM) for messaging, Friendster for online gaming, and hi5[4] for social networking, among others. However, with the dot-com boom onset, there was a rush to own infrastructure, consolidate independent Internet services, and control the network [83]. Consequently, many of the early OSNs, which were initially local and community-centric (hence decentralized), became global and were extremely centralized [31]. Significantly, many early OSNs are no longer in use or have grown to include more OSN services, sometimes by acquiring the smaller ones. The current breed of OSN applications has moved towards integrated services managed and controlled by a single entity, such as Facebook, YouTube, Twitter, and others. The initial centralized computing model utilized was the Client-Server model, making it easy to maintain and manage the data, the network peripherals, and network security. However, this model was disadvantageous because of unbalanced load distribution, presented a performance bottleneck, was a single point of failure, and a channel bottleneck [145]. Alternatives to the traditional Client-Server computing model developed to solve these challenges include cloud computing and operating own server farms. However, these do not eliminate the concerns and instead introduce others. To sustain the viability of OSNs on the centralized architectures, providers compel users to make certain compromises to use their services. These compromises have raised several concerns, some of which have moral and ethical implications. We give a brief analysis of the concerns raised and the proposed solutions to address them.

Concerns raised

The main concerns we identify in the centralized computing model fall into two broad categories, accumulated costs for centralized operations and security and privacy concerns [63], which we discuss further.

a) *Accumulated costs of centralization*: The accumulated costs, such as electricity, heating, ventilation, and air conditioning (HVAC), staffing, and system maintenance costs, among others, are a direct result of the linear, but massive, growth in the number of system users. Thus the challenges here are directly associated with the system scaling, such as handling a large number of connected users, infrastructural concerns, network traffic management, user-generated content (UGC) management, and database scalability [94]. Many of the services that most OSNs provide are free, automatically encouraging network growth. Thus, more resources are needed to run the services and improve them. The providers resort to user data monetization, which is possible as they have data sovereignty over both the user-generated content and the users' private information stored by the provider. By applying data analytics algorithms, the providers and other third parties gather information not inherent in the data. They can then utilize this to create a business advantage [112]. Now, most OSNs include a pay-to-play algorithm that allows third parties to embed advertisements into the user-generated content, specifically videos. This emerging trend leads to the second concern.

b) *Security and privacy concerns*: These are further analyzed based on the source of the threats, thus, user-related and provider-related [80, 117]. *User-related threats* may be intentional, such as by hackers, or unintentional due to users' poor privacy settings.

[2] https://myspace.com/

[3] https://www.icq.com/

[4] https://secure.hi5.com/

> With adequate security measures implemented in the OSN, the user-related threats will significantly reduce. The more significant concern is due to the provider. *Provider-related threats* emanate from the self-disclosure requirement during registration, which assumes a trustworthy service provider who will protect the user's private information. Because of this self-disclosure, the system providers can gather accurate data about the user activities while using the system and, based on the patterns, can predict the users' behavior. This data has opened up a new kind of market, called personal data markets [137], with the user as the commodity on sale. Third parties, who buy this data, use it to influence users' shopping habits through personalized advertising [137], emotional manipulation [9, 88], psychographic profiling [72], and online activity surveillance [10, 132, 140].

Due to these concerns, we identify several proposals to mitigate them, which we discuss next.

Proposed mitigating solutions

There have been various attempts to address these two challenges. One of the solutions proposed to address the cost of running the OSN is requesting donations like in Wikipedia. Another solution, which aims to reduce the infrastructural load, is to impose constraints on the type of data transmitted. Twitter, for example, imposes a limit on the number of characters and only allows short, lightweight video/audio clips. Also, to handle the challenges related to the monetization of the user-generated content and the user's private information, legal restrictions have been put in place. For example, within the European Union (EU), this is stipulated under the General Data Protection Regulation[5] (GDPR). These solutions only serve as stopgap measures, and further, they only focus on one concern at a time. Therefore, a singular, all-encompassing solution is needed to handle both problems effectively. With this goal in mind, [30] directs us to two ways to consider in the pursuit of such a solution: *extension of the capabilities of provided services* and *decentralization of supporting infrastructure*.

Extending the features and services focuses on what the application offers the users. Such extensions may require incorporating services that improve service delivery and guarantee security and access control. Such extensions have been ongoing since the inception of the early OSNs. These evolved from providing basic services like messaging only, such as in ICQ, or management of contacts as in Hi5, to become platforms that incorporate multiple services allowing third-party plugin applications. Centralized OSNs have largely achieved service extensions, but limitations arise as the OSN scales up [94]. Thus, pressure mounts to generate an income to sustain the infrastructure and cover maintenance costs and for-profit purposes, leading to monetization of user data [39]. The aspect of data sovereignty by an omnipotent service provider who may infringe the users' trust leads to privacy concerns [60, 120].

On the other hand, decentralization of the infrastructure remains open for further exploration. It entails dividing the administrative control at the back-end of the OSN so that no single entity has full control of the entire network. Therefore, computing models that support a decentralized online social network (DOSN) are worth consideration. These show the potential of giving users control over their privacy, more data security options, and data ownership, increasing system extensibility with fewer prohibitions, offer more reliability, and provide freedom of communication [143]. There are two general classes of decentralized online social networks (DOSNs), namely web-based and P2P-based [119], which we discuss further.

[5] https://gdpr-info.eu/

a) *Web-based DOSNs*: This is also commonly referred to as decentralized, federated OSN, for example, "Fediverse"[6]. Federated networks are formed by a collection of independently-hosted web servers that implement free and open-source software (FOSS), which support several communication protocols. An independent user can host a web server, called a node, which forms part of the federation. Users who desire to join the OSN then connect via a chosen "trusted" node. This setup significantly reduces the costs incurred due to centralization as several independently-hosted servers do the administration, but it does not eliminate this concern [142]. Although none of the nodes have a complete global view of the private data stored in the system, they still have access to some of the users' data. Therefore, the monetization problem still exists as a potential privacy threat.

b) *P2P-based DOSNs*: They utilize a fully decentralized model, in which each user simultaneously acts as a server and client. The users have full control over their data and support the network by providing the necessary infrastructure. They mitigate data privacy concerns by implementing suitable security mechanisms to prevent tampering with or reading the replicated data from other users without their permission. Thus, the P2P model promises to be a viable all-round solution for implementing DOSNs that address the concerns raised. However, to match the functionalities of the more popular centralized OSNs, P2P-based OSNs may require novel mechanisms to make them competitive. Meanwhile, additional overheads in terms of operational cost should not arise. We believe this is possible.

Availing functionalities and achieving the quality of service of the centralized OSNs, while addressing the concerns raised requires developing suitable P2P mechanisms to improve service delivery. Therefore the required OSN (functional and non-functional) requirements must be well understood to guide in describing an appropriate set of the technical requirements for a P2P-based OSN. We discuss this further in the next section.

2.1.2 A P2P Framework for Online Social Networks

In choosing to utilize the P2P model as the basis for the OSN, system developers need to identify the necessary application features. In this regard, the core aspects identified in centralized OSNs can inform the developers' design decisions. These features form a good template for what constitutes the user's expectations for the OSN. [81] gives a discussion of these features from which we derive the features that are relevant and present the summary.

- Identity - The extent to which users are willing to disclose their private details.
- Conversation - The ability of the users to communicate with each other.
- Sharing - The ability of users to exchange, distribute, and receive content.
- Presence - The ability to know of the availability and accessibility of other users.
- Relationships - The ability of users to establish social connections that lead to conversations, content sharing, and listing others as friends or followers.
- Groups - Ability of users to form communities and sub-communities.

With the desired set of OSN features clearly outlined, we can derive a general set of system requirements for OSN.

[6] https://the-federation.info/

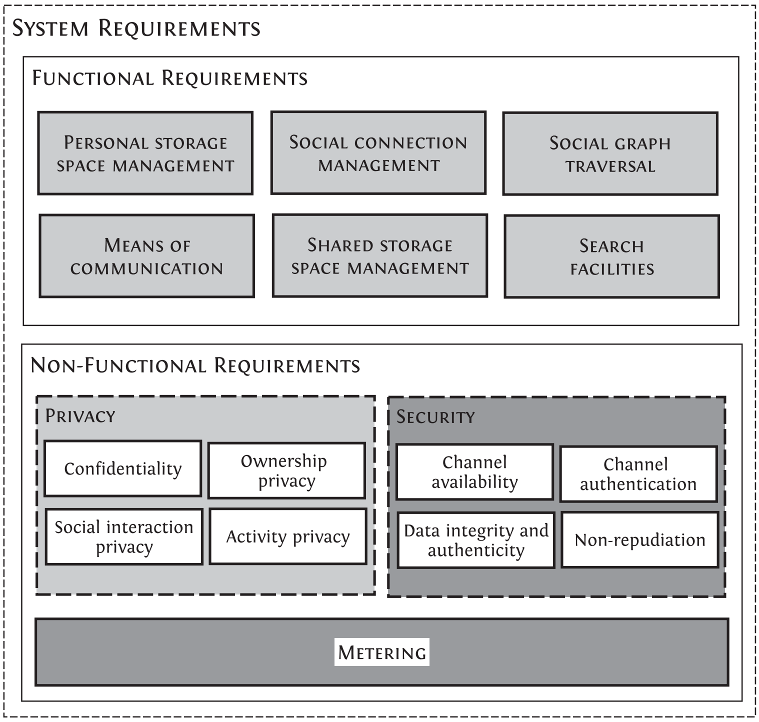


Figure 2.1: OSN system requirements

System requirements for OSNs

The system requirements define what the system must do to meet the users' needs. The two types of system requirements are functional and non-functional requirements, summarized in Figure 2.1.

a) Functional requirements: They specify what the system must do to achieve its main objectives [128]. They also explain some capability that one or several components provide [155] or a system's behavior under certain conditions [151]. The functional requirements identified are briefly presented [4, 156].

- Personal storage space management - After creating the OSN account and profile, he users should control some randomly assigned space to store, delete, and manipulate (or edit) their content.

- Social connection management - This gives users the ability to define how they relate to others by establishing, maintaining, or revoking a social connection, such as using a friend list.

- Social graph traversal - This supports traversing the online social graph and gives users the ability to examine other users' friend lists.

- Means of communication - This requires ensuring a secure channel for interactions between the users through messages such as text, audio, video and photos.

- Shared storage space interaction - This ensures that the users can interact via walls, forums, or commonly shared folders and form shared community spaces.
- Search facilities - This allows users to explore the social network space to find and connect to other users.

b) Non-functional requirements: These are necessary qualities that render the system attractive, usable, fast, reliable, and safe [128]. The requirements describe a property that the system exudes or a constraint that should be respected [151]. [4] presents privacy and security requirements, and we add metering as an additional requirement.

- Privacy - To guarantee this, the system developers should incorporate confidentiality, ownership privacy, social interaction privacy, and activity privacy.
- Security - This is made possible by incorporating channel availability, channel authentication, data integrity and authenticity, and non-repudiation.
- Metering - This allows the system developers and the users to collect statistics relating to the system's performance during operation via an integrated performance monitoring tool and use the statistics to improve the system's overall performance by making adjustments for optimality.

The identified system requirements now form a good basis for developing the technical requirements for a P2P framework for OSNs. We present this next.

Technical requirements for the P2P framework

Taking into account the functional and non-functional requirements, we now define the technical requirements as presented in Figure 2.2

a) *Overlay network*: This layer connects the application to the network layer by providing an interface that connects the system to the Internet.

- Secure routing - It should incorporate routing mechanisms for the efficient sending and receiving of messages. Also, the overlay should be able to handle churn, which leads to stale routes. The routing management scheme should identify them and replace them with alternative routes.
- Profile management - The user's profile is a data item that stores the user's login credentials and associated data. The overlay should include an addressing scheme for the nodes and data items by providing immutable `ID`s. It should also handle a change in the physical location of the users.

b) *The framework*: This layer includes all the OSNs application services needed for smooth functionality.

- Storage mechanisms - There should be support for simple files and complex data types such as forums, comments, and albums. It should have data redundancy using appropriate mechanisms such as replication and caching to guarantee data availability. It should also include appropriate searching mechanisms for locating and retrieving data.
- Identity control - The OSN should support group functionalities to allow users with common interests to interact and share data. To prevent other users from accessing

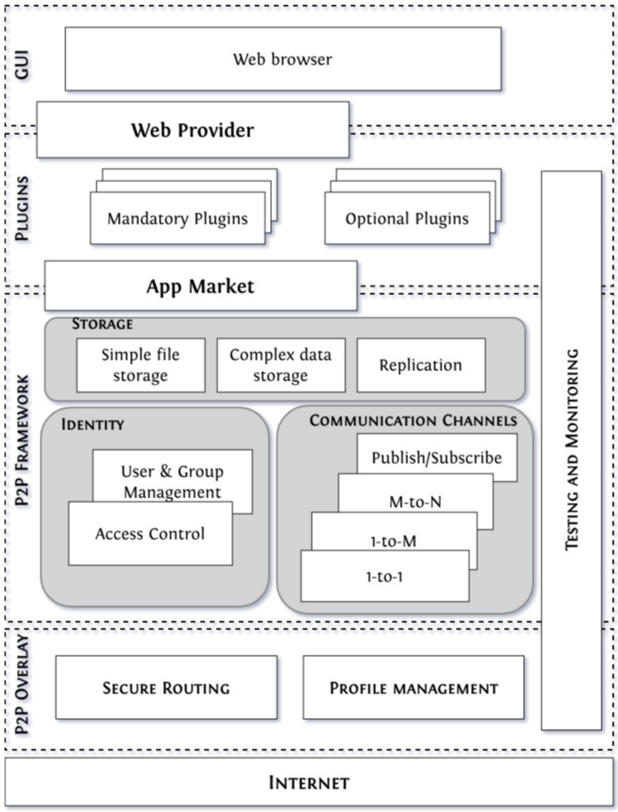


Figure 2.2: Architecture based on the technical requirements

unauthorized data, the OSN should include access control mechanisms for granting access rights.

- Communications channels - The system should aim at supporting direct (1-to-1) and multicast (1-to-M, N-to-1, and N-to-M) communications. It should also support asynchronously and synchronously communication. Another possible communication alternative to handle event-based notifications in the OSN is the publish/subscribe systems, which should also be incorporated.

- Quality monitoring and control loop - To learn more about the performance of the OSN, developers should include a monitoring layer. This layer should not interfere with the operations of the OSN and only collecting network statistics. By analyzing the collected statistics, information on the system's quality can help developers and users make appropriate adjustments to the application settings for performance improvement via a control loop.

c) *Application and graphical user interface (GUI)*: The application layer should include functionalities common to OSNs. It should also support additional functionalities with few modifications in the lower layers. For extensibility, developers should aim at a modular design with the OSN functions implemented as plugins. As most OSNs are web-based, the application should also be accessible via any commonly used web browser.

P2P technology has evolved over the years. Thus various options are available for each P2P component for use in meeting the stipulated technical requirements. To discover the optional mechanisms offered under each component, we survey the main components needed for a functional P2P OSN. We present a summary of our findings next.

Suitable P2P Components

Based on the proposed technical requirements in [98], we identify six significant P2P components that address all the requirements. We present a brief discussion of these components.

a) *Overlay*: An overlay is a new logical layer introduced on top of an existing network, which extends the underlying network (the underlay) by routing structures and forming connections between peers resulting in a new routing functionality [5]. The overlays proposed in the literature can be categorized into two main groups, which are *single-layer* and *multi-layer overlays* [85, 93]. Single-layer overlays employ only one overlay which handles routing and performs address management. They are further categorized based on the indexing mechanism used for locating nodes, shared resources, and groups, hence centralized, distributed, or hybrid, or the structure type, which looks at how the overlay performs routing, thus structured, such as Pastry [129], Chord [139], and Ca-Re-Chord [12], or unstructured such as GNUnet[7], Gnutella and Kazaa. Multi-layer overlays (also referred to as hierarchical overlays) combine the single-overlays to achieve desired functionality by either stacking them in a horizontal or vertical format. For horizontal multi-layer overlays, the overlays (each referred to as a leaf) join to form a single distributed hash table (DHT)-based P2P network. This design optimizes the routing and maintains the conceptual hierarchy of the leaf overlays. In vertical multi-layer overlays, the overlays cluster on top of each other, with communication taking place via gateway nodes. [85, 93] also discuss a third group, *bio-inspired overlays*, which implement algorithms mimicking biological phenomena such as swarm intelligence for routing functionality. We view this as an extension of the unstructured overlay.

b) *Search & lookup*: P2P systems should preferably make use of distributed indexing mechanisms. These mechanisms can either be *semantic-free*, *semantic-based*, [127] or *multi-dimensional indexing* [15, 157]. Structured overlays use semantic-free indexing (lookup) as the object `ID` is associated with a specific peer. They can, therefore, take advantage of the key-based routing (KBR) application programming interface (API) for data lookup. For a structured overlay network composed of N nodes, the lookup time is $O(logN)$. Semantic-based indexing (searching) is mostly associated with unstructured overlays that use flooding or probabilistic searching techniques. Developers can achieve multidimensional indexing in several ways. One method is to extend the P2P overlay by applying identifier space partitioning techniques. Examples include LobSter [6], which uses space-filling-curves to map two dimensional data to one dimension that can be maintained in the underlying DHT, *MAAN* [21], which uses a locality-preserving hash to map attribute values onto Chord, and *SCAN* [141], which augments CAN's overlay with long links based on Kleinberg's small world [82]. Another alternative is to combine P2P overlays with centralized approaches such as in *RT-CAN*, [147] which combines CAN and R-tree in a cloud system. Such modifications make it possible to perform multidimensional queries such as range queries, kNN queries, and window queries.

[7] https://gnunet.org/en/index.html

c) *Storage & redundancy management*: A reliable storage system is essential for any network. For P2P networks, the two main techniques for ensuring data availability are *replication* and *erasure codes* [22]. Several copies (replicas) of the same data item, either fully or partially, are stored at different network locations with replication. Replica management techniques are employed to guarantee file consistency. In many cases where replication is used, caching mechanisms are also included, aimed at reducing the overlay requests for data previously requested for and already availed in the local cache, which must guarantee data freshness (latest version of the data is cached). Error-correcting codes (ECCs) such as Reed-Solomon codes [125] proactively identify missing or corrupted encoded data portions in the case of erasure codes. Systems that store static data, such as file sharing applications, should preferably rely on erasure codes for data redundancy. Replication is suited for applications that store frequently updated data such as OSNs.

d) *Distributed data structures for complex data storage*: In OSNs, storage of some data items such as albums and comments in the simple file storage techniques such as PAST [35, 130] is not possible. Thus unique data structures that support nesting, such as distributed data structure (DDS) like sets, linked-lists and trees [2], need to be included. distributed data structures (DDSs) are distinguished based on how insertions and removals are performed. Thus we have *hash-based* or *order-preserving* DDSs. Hash-based DDSs such as DHTs rely on a hash table to map the keys to values. Therefore they do not need to maintain any order for the keys in the structure. Order-preserving DDSs strive to maintain key order during insertions and removals into two categories based on how they maintain the structure: *rotation-based*, such as AVL trees, and *split-and-join-based*, such as skip tree and skip list. However, the built-in structural repair mechanism to maintain the key order may introduce additional overhead.

e) *Communication*: The most commonly utilized communication models in P2P technology are *unicast* and *multicast* while avoiding broadcasting to minimize network traffic. Using unicast, functionalities such as direct messaging, video/audio chatting, and file sharing are possible. Multicasting is useful in collaborative initiatives such as group chats and updates to subscription channels. Multicasting can be one-to-many (1-to-M) such as in audio/video streaming applications, many-to-one (N-to-1) such as in resource discovery and polling system, or many-to-many (N-to-M) such as in multimedia conferencing and multi-player games [122]. Applications that offer scheduled audio/video streaming utilize 1-to-M multicasting. Another useful communication tool for P2P applications is the *publish/subscribe system*. This event-driven notification system makes it possible to distribute data/information by publishers (data/event producers) to subscribers (data/event consumers).

f) *Monitoring & management*: Monitoring the P2P network is needed to obtain in-depth statistical information using predefined metrics. Management relies on the monitoring data to inform changes to the system in case of poor system performance statistics. In P2P networks, monitoring approaches can be *unstructured* or *structured*. Unstructured approaches use gossip-based information exchange, such as T-MAN [74], push-sum [14] and continuous gossip-based aggregation [124]. Structured approaches build tree-like structures such as SkyEye [51, 54] and Mr. Tree [34]. Using the results gathered from the monitoring solutions, with a suitable control loop [58, 59], the system can automatically, or the users/developers manually make adjustments to get the desired performance.

Having looked at the functionalities available for each P2P component, we now shift focus to P2P-based OSNs. In the next section, we present a summary of proposed solutions identified

Table 2.1: Peer-to-peer based OSNs

Overlay Structure	Type	Proposal	Services	Status
Single-overlay distributed	Structured	LifeSocial.KOM/LibreSocial	Mixed services	Prototype
		Porkut/My3	Mixed services	-
		Megaphone	Microblogging	-
		eXO	Mixed services	-
		DECENT	Mixed services	-
		PESCA	Mixed services	-
		WebP2P	Mixed services	Prototype
	Unstructured	PAC'nPOST	Microblogging	-
Single-overlay hybrid	Structured	PeerSoN	Mixed services	-
		Safebook	Mixed services	-
		Cuckoo	Microblogging	Prototype
		HorNet	Microblogging	-
		LotusNet	Mixed services	-
	Unstructured	Litter	Microblogging	Prototype
		SuperNova	Mixed services	-
		Vegas	Microblogging	Prototype
		Tran et al.[144]	Mixed services	-
		HPOSN	Mixed services	-
Multi-overlay		Cachet	Mixed services	-
		Twister	Microblogging	Deployed
		DiDuSoNet	Mixed services	-
		SEDOSN	Mixed services	Prototype
		Blogracy	Microblogging	Prototype

in the literature and discussed in [98], paying particular attention to the components they use and the services they include.

2.1.3 Peer-to-Peer based OSNs

OSNs designed on P2P networks emerged as a solution to tackle the two concerns raised. Each proposed OSN implements the required functionalities using different P2P component mechanisms, applying novel techniques where necessary. 23 proposed P2P-based OSNs are surveyed, differentiated based on their services as either *microblogs* or *mixed services*. Many OSNs offer users a combination of direct messaging, content sharing, video conferencing, group interactions, among other services. We refer to these OSNs as mixed service OSNs. Microblogs, however, impose a size limit on the content users can share in a single post (text, pictures, links, short videos, or other forms of web media) and offer a limited number of OSN services. The OSNs are all placed into three categories, regardless of services that they offer, based on the overlay as shown in Table 2.1 and summarized below.

a) *Single-overlay distributed*: This group covers all OSNs that implement a single overlay. The group is further broken into two classes, structured or unstructured. OSNs identified as being structured are LibreSocial [55] (previously called LifeSocial.KOM [52, 53, 56, 57]), PorKut [109]/My3 [108], Megaphone [121], eXO [92], DECENT [73], PESCA [123], and WebP2P [33]. The only unstructured OSNs discussed is PAC'nPOST [8].

b) *Single-overlay hybrid*: OSNs in this category utilize a single overlay for routing but rely on a centralized mechanism to support index management. The OSNs can be either structured or unstructured based on the overlay. Structured solutions considered are PeerSoN [19], Safebook [28, 29], Cuckoo [153], HorNet [71], and LotusNet [4]. OSNs based on an unstructured overlay are Litter [79], SuperNova [136], Vegas [38], Tran et al. [144], and HPOSN [148].

c) *Multi-overlay*: As the name suggests, these OSNs implement more than one overlay to achieve efficient routing, index management, as well as storage management. OSNs in this group are Cachet [113], Twister [44, 45], DiDuSoNet [61], SEDOSN [40] and Blogracy [43].

In the following, we present our deductions based on a comparative analysis of the proposed P2P-based OSNs.

Comparing the P2P-based OSNs

In [98], we consider five aspects for comparison: the OSN requirements (functional and non-functional), system status, the developmental timeline, the essential P2P components incorporated, and the security considerations. We summarize our observations.

a) *OSN requirements and system status*: Of the six functional requirements we discuss in Section 2.1.2, all the proposals fulfill at least four of them, namely personal storage management, social connection management, means of communication, and search facilities. The other two functional requirements, social graph traversal and shared storage space interaction, are generally implemented by structured overlay-based proposals. It appears that implementing these two functions in structured overlay-based proposals, which support semantic-free lookup instead of semantic-based searching, is less challenging. The non-functional requirements of security and privacy are fully or almost fully seen in mixed service OSNs and are, to a large extent, not fully implemented in microblogs. Only LibreSocial achieves the metering requirement.

b) *System status*: Here, consideration is on the presence of a prototype or deployment of the proposal. Of the 23 proposals surveyed, there are only seven prototypes, of which three are mixed service OSNs (LibreSocial, WebP2P, and SEDOSN), four are microblogs (Cuckoo, Litter, Vegas, and Blogracy), and the deployed proposal is Twister. Interestingly, of the solutions with a prototype or deployed, two are single-overlay distributed, three are single-overlay hybrid OSNs, and three are multi-overlay OSNs. Of these, five proposals are microblogs. This revelation shows the ease and preference of developing microblogs instead of mixed service OSNs on the P2P platform. The additional fact that the singular solution deployed is a microblog strengthens this observation.

c) *Developmental timelines*: The development of the proposed P2P OSNs occurred between 2008 and 2016. The climax of the development is seen from 2012 to 2015, with 14 proposals developed over this period. We note that all the proposals are research projects in higher learning institutions (specifically in university environments).

d) *Essential components incorporated*: The surveyed proposals do not all implement the functionalities in the same manner. Many novel techniques are applied to implement the OSN functionalities, which shows the diverse nature of P2P components in being adapted and modified to realize the desired OSN function. This diversity is evident when considering the single-overlay hybrid and multi-overlay proposals.

e) *Security consideration*: Emphasis is on how the proposals handle identity management, access control, confidentiality, integrity, and anonymity. All proposals include identity management mechanisms to handle identity creation and verification. Microblogs do not to implement access control, confidentiality, and anonymity. Thus, microblogs do not guarantee activity privacy, social interaction privacy, and sometimes confidentiality. We further observe that proposals not paying particular attention to access control, confidentiality, integrity, and anonymity focused on a novel mechanism for improved service delivery or performance. Except for WebP2P, which does not incorporate anonymity, the other mixed service OSNs with prototypes addressed all security aspects. For the deployed microblogs, only Vegas and Blogracy included all the security features. Twister, the only deployed microblog, does not address access control and anonymity.

2.2 Contribution

We present the contributions made by this article to the body of knowledge in the following. First, we focus on the social networking aspects in general. We present the common social networking classifications based on their usage, data management design, system design, and network orientation. We then give an analytical discussion pointing to the need to consider other alternatives to the centralized/client-server data model because of the concerns related to maintenance costs and data privacy, leading to a fully decentralized data model, specifically the P2P data model.

Secondly, we analyzed literature for information on the desired features for an OSN to construct a suitable set of OSN system (functional and non-functional) design requirements. We then use these system design requirements to formulate the technical requirements for a generic P2P framework for OSNs and define the core P2P components for a suitable software architecture.

Thirdly, we consider each of the core P2P components in the software architecture and discuss the optional mechanisms available as presented in the literature. In this case, the goal is to show the developers' diversity of choice and present arguments for and against each specific component's mechanisms.

Finally, we discuss several P2P-based OSNs proposals. We compare and contrast them, considering the different P2P components they use to achieve functionality and synergize. The comparison focuses on the system design requirements included and the technical requirements they meet to realize a working application.

The article seeks to point to the use of P2P technology in OSN development as a plausible alternative for elimination of infrastructural and maintenance costs, guaranteeing users control of their privacy, ensure data ownership remains with the users, and offer a censorship-resistant platform.

2.3 Personal Contribution

Newton Masinde, the author of this thesis, contributed to this paper by undertaking background research of the various aspects of P2P technology and OSN. He also compiled the findings for most of the sections of the article. Kalman Graffi contributed through discussions on the structure and classification of the P2P technologies, and provided direction and timely criticism on the structure and content of the paper. He also wrote the section on the monitoring and management solutions and edited and proofread the paper.

2.4 Importance and Impact on Thesis

In this article, we seek to address several research questions. We first describe OSNs and their functions and discuss the implications of using the different architectural models (centralized, federated, and P2P). To address the research question RQ1.1, we identify the main OSN features typical to the popular OSNs. These features allow us to enumerate the system design (functional and non-functional) requirements for OSNs, thereby addressing research question RQ1.2. We handle the research question RQ2.1 by transforming the system design requirements into a list of technical requirements that define the core components anticipated for a P2P framework for OSNs. These P2P components are incorporated into a software architecture. We handle the research question RQ2.2 by presenting a targeted study on the various mechanisms available for each P2P component and give the pros and cons for each mechanism. Finally, to answer the research question RQ2.3, we revisit several proposed P2P-based OSNs to see how they address the various technical requirements. Of the 23 P2P OSNs, 8 were microblogs, and 15 offered mixed services. 5 out of the 8 proposed microblogs have implementations, with one currently deployed. In the case of the mixed services, only 3 have prototypes. The rest of the proposals are merely discussed on paper. Our P2P-based framework for OSN, LibreSocial, presented in Chapter 3, closely fits the stated requirements, and Chapter 4 presents an analysis of its performance. LibreSocial, however, still has some weaknesses which we have identified. First, the DDSs which implement complex data do not include a functional metadata-based search and retrieval mechanism. Hence, we address this in Chapter 5. Secondly, we observe that LibreSocial does not use social data to improve data availability for faster data retrievals. We, therefore, present a social caching mechanisms in Chapter 6. Lastly, we identify the challenge of handling heterogeneous nodes with different capacities that impact the network differently, which we call the weak/strong node support. In Chapter 7, we discuss a capacity management protocol for LibreSocial that tackles this challenge.

Chapter 3

LibreSocial: A Peer-to-Peer Framework for Online Social Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [55].

Kalman Graffi and Newton Masinde. "LibreSocial: A peer-to-peer framework for online social networks". In: *Concurrency and Computation: Practice and Experience* (Dec. 2020). SPECIAL ISSUE PAPER, pp. 1–26. DOI: 10.1002/cpe.6150

In this chapter, we present LibreSocial, our P2P framework for online social network (OSN) that meets the defined technical requirements to realize the four-fold software architecture. In the following sections, we give a summary of our paper in Section 3.1, present the contributions the paper makes to the body of knowledge in Section 3.2, discuss personal contributions to the work in Section 3.3 and give the importance and impact of the paper on this thesis in Section 3.4.

3.1 Paper Summary

This article presents the current iteration of a LibreSocial, our peer-to-peer (P2P) framework for online social network (OSN), previously called LifeSocial.KOM [52, 53, 56, 57]. LibreSocial's architectural design borrows heavily from the Open Services Gateway Initiative (OSGi™) service platform, allowing users to design large scale modular applications in Java. The application's design follows the software architecture presented in Figure 2.2 that fulfills the technical requirements discussed in Chapter 2.1.2. LibreSocial is composed of four distinct layers, *a) the P2P overlay*, *b) the P2P framework*, *c) plugins & applications*, and *d) the graphical user interface (GUI)*. We describe these four layers in further detail in the sections that follow. Furthermore, we give a brief discussion of the performance characteristics of the system.

3.1.1 The overlay: a heavily modified FreePastry

The overlay of LibreSocial is based on an open-source implementation of Pastry [129] called FreePastry[1]. This choice of FreePastry is because it provides efficient key-based routing and includes additional functionalities such as PAST [35, 130] (a storage and replication scheme), Scribe [131] (a simple multicast event notification scheme), and Splitstream [24] (a multicast streaming scheme that relies on Scribe).

[1] http://www.freepastry.org/FreePastry/

The overlay provides a circular `ID` space of size 2^{160}, and each node has a 160-bit `nodeID`, with guaranteed uniform distribution of the `nodeIDs` in the `ID` space. The overlay ensures efficient routing. Given a numeric value within the 160-bit numeric space and a message, FreePastry can perform message routing in $O(log_2 N)$ steps to a node whose `ID` is numerically closest to the given numeric value. Significant modifications have been made in FreePastry to include additional features that support a functional OSN. These are enumerated and described.

a) *Secure `nodeID`*: Elliptic curve cryptography [84, 105] (ECC) with 160-bit keys supports asymmetric encryption algorithm with the 160-bit public key as the `nodeID`. Therefore, communications can be encrypted and signed.

b) *Parallel and iterative routing*: The routing table was extended so that, instead of having a single entry per route, a bucket contains multiple addresses, similar to the Kademlia [103] DHT. Thus a requesting node sends messages to k different nodes in parallel and waits to receive feedback from all paths before iterating the process with a new set of nodes based on the replies received. Thus the routing process is significantly reduced, and we can mitigate the effects of wrong routes due to malicious nodes.

c) *Support for weak nodes*: FreePastry assumes that all participating nodes have equal computing capabilities, which is not the case in reality. Some nodes will possibly have less storage capacity, limited bandwidth, or have a shorter participation duration. Such nodes are considered *weak*, and the goal is to avoid routing through them but ensure they inclusion in the leafset for ease of communication with other nodes. This may be possible through the introduction of capacity-aware storage indirection as described in [150].

3.1.2 The P2P framework

The peer-to-peer (P2P) framework is a layer containing different P2P components that provide services to the application layer. These services include storage and replication, access control mechanisms, search and lookup mechanisms, distributed data structure (DDS) for complex data storage, secure communication functionalities, user and group management functionality, and a quality control loop via a testing and control mechanism. A description of the core components of the framework follows.

a) *Storage and replication*: LibreSocial uses PAST [129] for simple file management. The files are addressed using 60-bit identifiers and can be accessed using FreePastry's routing algorithm. Traditionally, PAST only provides INSERT and REQUEST file operations. In LibreSocial, PAST was modified to include UPDATE and DELETE file operations. PAST also supports data redundancy based on replication, with additional functionality for data caching, storage load balancing, and traffic load balancing.

b) *Access control*: Mechanism for read and write access are included based on symmetric cryptography. The data owner encrypts the stored item and the symmetric key for read access using each user's or the group's public key to allow read access. File updating (or overwriting) is possible if the corresponding private key used to sign the item belongs to a specified public key.

c) *Distributed data structures (DDSs)*: To meet the need for storing complex data objects such as albums and comment, which include nested data that is linked, LibreSocial includes *distributed data sets* to manage friends, uploaded files, and albums, *distributed* *linked lists* to manage comment sequences, forums, and inbox messages, and *prefix hash trees* for performing range-based searches.

d) *Communication channels*: LibreSocial supports sending synchronous and asynchronous messages using unicast or multicast communication channels. There is also an aggregation channel that is useful during the monitoring process for statistical aggregation. Other mechanisms include publish/subscribe (pub/sub) channel via Scribe, streaming via Splitstream to distribute the streaming workload among the participating nodes equally, and streaming via WebRTC[2] for simple low-latency audio/video conferencing. Using the public key as the `nodeID` in the network makes it possible to guarantee a secure message channel by message encryption (symmetric or asymmetric depending on the message channel). Signed messages can be easily verified using the `nodeID`, which is the node's public key.

e) *Monitoring and testing*: Using the OSGi™ design framework, a quality control mechanism that incorporates two components, monitoring and testing plugins, is possible. The testing plugin makes it possible to run tests that simulate user interaction via a command-line interface. In contrast, the monitoring plugin collects the system statistics at runtime. The monitoring component is designed based on a tree-based monitoring algorithm called SkyEye [51, 54].

f) *AppStore*: This is a plugin extension for the management of repository-based apps/plugins. It allows users to create, manage, and publish their plugin extensions and is similar to apt-repositories in Gnu/Linux, and other users can download and install the published plugins.

g) *Other supporting components*: The components in the P2P framework interact with one another and with other layers via three modules. The first is the *StorageDispatcher*, which provides storage services for storing data items locally and remotely. The second component is the *MessageDispatcher*, which handles message delivery via a MessageChannel, topic message delivery via a TopicChannel, and aggregation data message delivery via an AggregationChannel. The last component is the *Information Cache*, which acts as a local cache for objects or messages requested by the application layer plugins.

3.1.3 Plugins and applications

OSGi™ defines the different software components that add desired functionalities as plugins. In essence, these plugins are Java ARchive (JAR) files that can be added or removed at will. By designing several plugins that can interact, it is possible to develop a complex application such as LibreSocial. Therefore, developers can create any other application to use the overlay and framework, with some modifications as required. Table 3.1 shows the application plugins in LibreSocial.

3.1.4 The graphical user interface

LibreSocial's frontend uses standard Web technologies such as HTML5, AJAX, CSS, JQuery, Bootstrap, and Knockback.js. It includes multi-language support (currently only German and English) and can therefore run on any standard browser. The backend, on the other hand, is composed of three parts.

[2] https://webrtc.org/

Table 3.1: LibreSocial Plugins and their functions

Plugin	**Function**
Login	Entry point into the network. Handles the user registration and login processes.
Profile	Used to create a data item containing the user's information and data, which is stored in the network storage.
Notifications	Informs user about any new events within the network such as a friendship request or an invitation to chat.
Files	Allows users to store and distribute larger network files in the form of chunks.
Search	Gives the user ability to search for people registered in the network.
Friends	Stores relationships existing between users by forming the social interconnections that may be different from the actual network structure.
Groups/Forum	Models a community environment. Group users are provided with a shared drive for file storage and a forum for posting discussions.
Calendar	Allows saving of appointments and events chronologically.
Message	Similar to an email application. Allows users to send messages to and receive messages from other users.
Multichat	Gives users the ability to chat with one another. Supports conversations between multiple users.
Audio/Video chat	Provides 1-to-1 audio and video communication.
Wall	Provides 1-to-N communication. Users can view personal wall pages of friends and can post entries and comment on posts made.
Photos	Allows sharing of photos and photo albums with friends as well as comments on the photos.
Voting	Gives users the ability to conduct surveys among a predefined group of users or the entire set of users of the network.
Test	Used to test the performance and reliability of the other plugins.
Monitoring	Performs monitoring of the entire system, that is, the plugins, the framework and the overlay. Works in conjunction with the test plugin.
Error Console	Displays information generated by the framework while the user is logged in such as errors, warnings, and debug messages.

a) *The plugin template*: This is composed of HTML files that use Knockout.js[3], a standalone JavaScript implementation based on the Model-View-View-Model (MVVM) configuration.

b) *Plugin logic*: It oversees the process of user event transference from the frontend to the REST Handler via the WebProvider and back to the frontend, all the while rendering the data it receives to the required template.

c) *WebProvider*: This is a plugin acting as the interface between the application plugins and the web browser. It uses an embedded web server jetty to communicate with the browser. The jetty module handles the provision of static files, the REST interface, and the WebSockets.

It is important to note that LibreSocial, as an application, is not designed to run on mobile devices, mainly due to limits on resources such as bandwidth and availability. Moreover, [91] notes that many Internet Service Providers are not very friendly to P2P traffic. Neverthe-

[3] https://knockoutjs.com/

less, the GUI's design is mobile-ready and it is possible to tunnel the GUI for an instance of LibreSocial running on a desktop to a mobile device. Connections are encrypted using the normal Hypertext Transfer Protocol Secure (HTTPS) rather than Hypertext Transfer Protocol (HTTP) to ensure security between application front and back-end.

3.1.5 Performance characteristics

We introduce a baseline test workload to LibreSocial to obtain the cost of running it in terms of *network*, *storage*, and *security*. With 64 nodes forming the ring, network data rates oscillate between 0 and 150 MB/s. The network message rates oscillate between 0 to 6000 messages/s, and the hop count is at a maximum of 1 due to the small size of the network. The peak storage load for a single node is roughly 4100 unique items, with replicated items being approximately 14000. The overheads in terms of encryption and decryption of data items are relatively small as most objects are less than 1 kilobyte and encryption/decryption times are less than 1 ms.

3.2 Contribution

The contributions of this paper to the thesis are as follows. First, we revisit the technical requirements for a P2P framework for OSNs described in [98]. After that, we describe an OSN application called LibreSocial designed in line with the technical specifications and incorporating different P2P components such as those described in [98] to realize a working application. Lastly, we conduct a simple test on LibreSocial to review the application's performance and cost characteristics, the framework, and the network under load.

3.3 Personal Contribution

Newton Masinde, the author of this thesis, compiled all the material on the iterative development of LibreSocial up to the current status, and took over as the lead for its development from 2017. He also analyzed the documents to highlight the developmental changes between LifeSocial.KOM and LibreSocial. Further, he designed the experimental setup, performed the experimental work that showed the system's performance, and wrote most of the paper. Kalman Graffi provided LibreSocial and all the material on its developmental progression. He proposed and refined the design of LibreSocial, as described in [55]. He also provided critical feedback during the paper writing process, making necessary adaptions to its final form. The current iteration of LibreSocial acts as the foundation for the improvements described in Chapters 5, 6 and 7.

3.4 Importance and Impact on Thesis

With this paper, we fully address research question RQ2.4. We show that the proposed software architecture presented in Chapter 2 can be translated into a working OSN. We herein describe how the different P2P components are integrated into an OSN that meets the need for privacy of information and user's data and secure communications in a zero-trust environment, but also has the promise of being scalable and reliable at a low cost.

Chapter 4

Systematic Evaluation of a Peer-to-Peer Framework for Online Social Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [100].

Newton Masinde, Liat Khitman, Iakov Dlikman, and Kalman Graffi. "Systematic Evaluation of LibreSocial–A Peer-to-Peer Framework for Online Social Networks". In: *Future Internet* 12.9 (2020), p. 140. DOI: 10.3390/fi12090140

In this chapter, we present a systematic performance evaluation of LibreSocial. The rest of the chapter proceeds as follows. In Section 4.1, we present a summary of the paper. We give the contributions of this paper in Section 4.2, and the personal contributions in Section 4.3. Finally, we present the importance and impact on the thesis in Section 4.4.

4.1 Paper Summary

In this article, we present the results of undertaking systematic load testing of LibreSocial, our peer-to-peer (P2P) framework for online social networks (OSNs) presented in Chapter 3. In experimental testing of large scale distributed applications, we distinguish between the application under test and the testing environment. Consequently, there are four distinct experimental methodologies for large-scale system testing can be derived, that is, in-situ, emulation, simulation, and benchmarking [20, 65]. In in-situ experimentation, the real application is executed on real hardware and under real scaling conditions, such as in Planet-Lab[1]. For emulations, the tester creates a set of synthetic experimental conditions to execute a real application using virtualization or controlling the environment through performance degradation of the hardware such as the CPU or network speed. Simulations focus on a chosen part of the environment while the rest is abstracted. Thus execution of a model of the application executed. Finally, in benchmarking, a synthetic or generic application is executed in the real environment.

The experimental methodology we use to test LibreSocial is benchmarking because LibreSocial's design allows us to meet benchmarking's basic requirements. These requirements include creating a workload representative of the real-world application, exercising all the critical services, not involving tuning/optimization for testing, ease of reproducing the results, and no inherent scalability limitations anticipated [89]. P2P benchmarks need to clearly define the underlay's relevant properties to ensure reproducibility to differentiate them from standard benchmarks [89]. In general, a benchmark set can be viewed as a tuple consisting of quality

[1] https://www.planet-lab.org/

attributes Q, metrics M, and test scenarios S [86]. We define the benchmark set for our experiment shortly. Benchmarks are characterized by three important parameters [133], which affect the benchmark tuple in various ways. These parameters are:

a) *System parameters*: These are system-specific settings that bind the system during the test. These include the replication factor (RF), the leafset size, and the routing table size.

b) *Workload parameters*: These parameters influence the workload generation process during the testing. Such parameters include the number of peers and the application-specific settings such as the number of times a particular activity is carried out, like number of messages sent.

c) *Environment parameters*: These are usually not easily changeable as they are bound up in the host computer system, for example, storage capacity, memory capacity, data rate, and so on, as well as the underlying communication system.

4.1.1 Quality metrics & properties

In any system test, the metrics are useful only in describing a single attribute of any mechanism within a test scenario, workload, or configuration. On the other hand, quality properties consider the overall system's or a single mechanism's characteristics, defined using a set of carefully chosen quality metrics. Saller et al. [133] divide the quality properties into two groups workload-independent and workload-dependent. In the following, we describe the two types of quality properties briefly and highlight the properties, with the corresponding metrics, chosen for benchmarking LibreSocial with the relevant metrics.

Workload-independent properties These properties provide a measure of the system's behavior under test under differing workload conditions. For the benchmarking of LibreSocial, the main properties of concern are performance and cost, we further describe.

a) *Performance*: This is measured by taking matching the system's reaction time (*responsiveness*) and the system's load handling capabilities (*throughput*) against the expected. For performance measurements, metrics selected are *hop count*, *storage and retrieval times*, and *message sending & receiving rates.*

b) *Cost*: Here, the focus is on the resources consumed or utilized as a result of fulfilling a given task or providing a needed service. The metrics identified are *used storage space*, *used memory*, and *network bandwidth.*

Workload-dependent properties The properties in this category are affected by the workload introduced into the system under test. With these properties, it is possible to redefine the workload to achieve a desired effect on the overall system performance. The properties selected for evaluation are stability and scalability, which we elaborate on further.

a) *Stability*: It describes the system's ability to continue operating despite inherent system behavior dynamics such as churn, eventually converging into a stable state. The relevant metrics identified are the *number of network nodes*, *leafset size*, and *routing table size.*

Table 4.1: Plugins and plugin actions

Plugin	Action
Messaging	Send message View inbox message View outbox message
Live chat	Send multichat invitation Send multichat message
Group	Create group Invite friend to group View group View my group list
Filestorage	Create folder Upload file in folder View folder
Photos	Create photo album Upload photo View own album View friend's album

Plugin	Action
Forum	Create forum thread Comment forum thread View forum
Calendar	Create calendar event Edit calendar event View calendar
Voting	Create vote Add public vote Voting invite user Vote Get my votings Get voting results
Wall	Send wall post Comment wall post View own wall View friend's wall

b) *Scalability*: This is concerned with the overall system behavior under changing workload conditions. Two scaling dimensions to consider are *horizontal*, which considers the number of peers in the network, and *vertical*, which looks at the peers' actual workload. In the testing, we focus on horizontal scalability, measured using the same metrics as stability.

4.1.2 Experimental environment and test scenarios

The test environment we use is the high-performance computing (HPC) cluster provided by the "Centre for Information and Media Technology" (ZIM) at Heinrich Heine University. We select four test scenarios to assist in making conclusive remarks on the quality of service (QoS) of LibreSocial. These scenarios are *plugin analysis*, *pseudo-random behavior*, *scalability/stability*, and *replication factor*, which we discuss in the following.

Baseline tests–Plugin analysis These tests function as the baseline tests for the system. As the OSN layer in LibreSocial is based on plugins (Java bundles implementing different OSN functionalities), these tests focus on showing the overall effect that each plugin (hence OSN function) has on the system in totality. Table 4.1 shows the plugins and their associated actions. A plugin action is the result of a particular activity initiated by the user. We employ two different network sizes during the testing, 100 and 500 nodes. For each, we initiate two different workload sets, specifically, light load (two repetitions per plugin action) and medium load (five repetitions per plugin action). For each plugin action, the repetitions are allowed to complete over a five minute duration.

Pseudo-random behavior The goal of these tests is to mimic users' behavior in the real world by applying a user behavior model based on the work done in [11]. The test is conducted for 500 nodes as follows. In the first 160 minutes, all the nodes join the network. After that, the nodes execute the plugin actions based on the user behavior algorithm for 200 minutes.

Scalability/stability In this scenario, the focus is on testing the network's response to growth and churn. Therefore, the test is conducted in two phases, growth and churn. During the growth phase, the network increases in steps of 250 nodes to reach a maximum of 1000 nodes. We initially begin with 250 nodes. The growth steps represent an increment of double, 1/2, and 1/3 of the network's total nodes. We initiate churn when the network size reaches 1000 nodes by removing 250 nodes from the network in three steps, representing 1/4, 1/3 and 1/2 of the total network nodes in the network, respectively, until we remain with a network consisting of 250 nodes. We execute the same workload on the nodes at each step of either network growth or network churn.

Replication factor The value of the replication factor (RF) in the system settings ensures that the appropriate number of replicas is created when a data item is stored to ensure data persistence even in the absence of the data owner. The value for the RF in FreePastry is directly related to the leafset size based on Equation 4.1. We conduct tests for 100 and 200 nodes using RF 4, 16, and 32, and 1000 and 2000 nodes with RF of 4 and 16. This test aims to see the effect of the replication factor on storage performance in general and the associated costs.

$$Replication\ Factor = \frac{Leafset\ size}{2} + 1 \tag{4.1}$$

4.1.3 Summary of findings

In the following, we give a summary of our deductions based on each of the test scenarios.

Plugin analysis We observe that the OSN plugins synergize exceptionally well with the framework and overlay. Our analysis of the plugin actions revealed three impact groups:

a) *Low network messaging and low data rates* - The overall effect causes less than 1000 network messages/sec and less than 10 Kb/s. The plugin actions in this category are mainly those that retrieve data for purposes of viewing.

b) *High messaging and low data rates* - Here, the overall effect of the load causes 100 to 15000 messages/sec and less than 15 Kb/s. The plugin actions in this category are mostly those that utilize the distributed data structures.

c) *High messaging and high data rates* - The load causes generation of 500 to 23000 messages/sec and more than 15 Kb/s. Plugin actions in this category are primarily those that work in conjunction with the Filestorage plugin to store and retrieve data.

Therefore, we make the observation that the different actions performed by the plugins have varying effects on the entire system, with the most significant effect being due to storage and retrieval of files by the Filestorage plugin.

Pseudo-random behavior Under pseudo-realistic conditions, the system remains stable throughout the test duration, which we view as a testament to the system's good response and performance under randomly initiated user actions. However, we note a challenge with the behavior algorithm applied and not the application itself, which we deduce from the declining node count. We had not anticipated this, and we believe that this may necessitate further investigation into how the behavior algorithm interacts with the plugins.

Stability and scalability We observe that the system scales easily, up to at least 1000 nodes, and during the process of churn, the system adjusts itself to accommodate the sudden departure of many nodes.We, therefore, conclude that the system is stable even under churn conditions, and is also scalable.

Replication factor Finally, under changing RF, we see that higher RF causes an increase in network costs incurred (storage, memory, and bandwidth). However, there is an improvement in the average performance, especially when the RF is higher in a network with significantly more nodes. Also, when comparing the maximum recorded values for the performance and cost metrics, we note that they increase with increasing RF and network size. An interesting observation is a non-linear effect on the performance when the network size increases while the RF is kept constant. This observation leads us to conclude that as the network size grows, it is prudent to ensure the value of the RF also increases to meet the needs of the growing network and maintain optimal storage performance.

4.2 Contribution

Aside from the overall comprehensive goal of evaluating the system in general as the main contribution from this work in meeting this thesis's core objectives, other vital contributions are as follows. We describe the benchmarking process for LibreSocial, clearly explaining the core quality properties with their relevant metrics and the test scenarios based on acceptable testing standards. We contrast the methodology applied for testing LibreSocial with other discussions on the performance analysis of proposed P2P OSNs presented in the literature, most of which apply simulation testing. Our experimental methodology, benchmarking, ensured that we could test LibreSocial without introducing biases through optimization or performance tuning. Thus the results we obtain are very close to what may be very similar to an in-situ experiment.

4.3 Personal Contribution

The contributions of the authors are as follows. The work's conceptualization was by Newton Masinde and Kalman Graffi. Newton Masinde, Liat Khitman, Iakov Dlikman, and Kalman Graffi defined the quality properties and evaluation metrics. Newton Masinde, Liat Khitman, Iakov Dlikman, and Kalman Graffi worked on the monitoring setup, definitions for the workload generator, workload compositions, and definitions (setting of various parameters and events frequencies). Liat Khitman did a partial redesign and debugging of the test plugin for LibreSocial. Computational support and infrastructure needed for the setup of the testing environment were provided by the "Centre for Information and Media Technology" (ZIM) at the Heinrich Heine University (Germany). Newton Masinde did the experimental work for the plugin analysis-baseline test (initially done by Iakov Dlikman during his Master thesis). Liat Khitman did the experimental work for pseudo-random behavior, scalability/stability, and replication (as part of her Bachelor thesis). The results' analysis included efforts by Newton

Masinde, Liat Khitman, Iakov Dlikman, and Kalman Graffi. Finally, Newton Masinde, Liat Khitman, and Kalman Graffi wrote, reviewed, and edited the paper before publication.

4.4 Importance and Impact on Thesis

System testing is an integral part of ascertaining the quality of service (QoS) of any system that is under development. In Chapter 3, we presented LibreSocial, our P2P framework for OSNs. The conducted performance analysis was only a small component of [55], aimed at showing a working application. The goal of this article is to meet the questions raised under the problem statement PS4 fully. To answer the research question RQ4.1, we conduct action-based analysis tests to find out the performance due to the actions. To address the research question RQ4.2, we apply a pseudo-random behavior model to mimic actual users and assess the impact on the network. By performing step-wise upscaling and downscaling to the system to simulate network growth with loading and network churn with loading, we address the research question RQ4.3 and show that the system scales easily, handles churn well, and remains stable through the entire test. Finally, as an answer to research question RQ4.4, which gives us more insight into how to improve storage performance in the network. We view the evaluation presented in [100] as a first, primarily because of the experimental methodology. The tests demonstrate the possibility of achieving a working OSN that offers a wide range of functions to the users in a P2P environment.

Chapter 5

Search Algorithms for Distributed Data Structures in P2P Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [1].

Raed Al-Aaridhi, Iakov Dlikman, Newton Masinde, and Kalman Graffi. “Search Algorithms for Distributed Data Structures in P2P Networks”. In: *Proceedings of the 5th International Symposium on Networks, Computers and Communications (ISNCC 2018)*. IEEE, June 2018, pp. 1–8. DOI: 10.1109/ISNCC.2018.8530977
Acceptance Rate 38%

In this chapter, we present the first of three proposed mechanisms to improve service delivery, search algorithms for metadata-based searches in distributed data structure (DDS). The chapter is organized as follows. In Section 5.1, we present a summary of the paper. After that, we give the contribution of the paper in Section 5.2 and the personal contributions in Section 5.3. Finally, we give the importance and impact on the thesis in Section 5.4.

5.1 Paper Summary

The main task in this paper is to handle the problem of improving the searching within a distributed system, and in particular, a peer-to-peer (P2P) environment that utilizes distributed data structure (DDS). Efficient data retrieval in distributed applications must include an efficient search algorithm to locate the needed resource, either the node or a data item. One proposal is to use the metadata, the data or information about the files or documents stored in the system. For this to be possible, the objects/items features reveal what they contain or are about (content), give details of the object features (context), and provide a formal set of associations within or among the individual objects [47].

As an example, consider a query posed to a phone seller. Searchable items would be the mobile phone. The possible attributes in the search query can include *Model*, *Model_ Number*, *Price*, and *Year*. These attributes should not necessarily be statically defined and can be bound by a relation such as *smaller than*, *greater than*, *between*, or *an exact match*. A possible query may specify the attributes expressed by Predicates 5.1, 5.2, and 5.3 to be met. Scheme 5.4 is a possible regular expression of a search query. Thus, exact-keyword searches, partial answer queries, range queries, partial and similarity matches, and k-Nearest neighbor searches are possible.

$$Model = \text{“}\texttt{Samsung}\text{”} \tag{5.1}$$

$$Model\_Number \in \text{“}\texttt{S10, S11}\text{”} \tag{5.2}$$

$$Price < \text{“}\texttt{1000€}\text{”} \tag{5.3}$$

$$Query : (Keyword|(Attribute1 = Value1, ..., AttributeN = ValueN)) \tag{5.4}$$

Many P2P applications utilize either of two types of distributed indexing mechanisms, semantic or semantic-free. *Semantic indexes* are more common in unstructured P2P networks and work well with metadata-based searches. On the other hand, *semantic-free indexes* are more common in structured P2P networks that utilize key-based routing (KBR), such as distributed hash tables (DHTs), hence are useful in exact match queries but poor for metadata-based searches. The task of searching and retrieval in a distributed network environment generally proceeds in three steps [76, 77]: *a*) identifying the needed resource using its metadata (*metadata search*), *b*) verification of the actual resource's location (*document location search*), and, *c*) eventual downloading of the identified resource (*document download*). However, most P2P applications are designed on structured networks that rely on semantic-free indexing techniques for data retrieval. Therefore, the challenge is to develop a suitable metadata query engine that indexes using a set of keywords (stored as metadata) and retrieves the document or documents while also taking advantage of the existing key-based search mechanism. Further, we desire that the query engine design be overlay-independent to support scaling, heterogeneity, and load distribution. We consider the design challenges associated with meeting this goal.

5.1.1 Design challenges

The design challenges that this paper seeks to tackle are: appropriate storage techniques to support the use of keyword-based queries for document search, mechanisms for performing document searching, and the join algorithms to consolidate the retrieved document to match the data query. We discuss these challenges in the following section.

Storage techniques for keyword-based searches

To support keyword-based queries in structured P2P applications, redesigning the storage strategy to handle storage indirection is necessary. In storage indirection, a pointer is used to access the reference to an object. The reference in our case is the keyword or set of keywords. Three storage strategies are possible, local, direct, and indirect through referencing [18], with the summary of their features shown in Table 5.1.

a) *Local storage* - The indexing peer keeps the documents along with their keywords locally. The keywords are stored in the DHT and associated with the indexing peer's address, which points to itself. The pointers are organized in publish/subscribe trees. Queries are sent to a single node responsible for the keyword and then forwarded to nodes with the pertinent documents that evaluate the query locally and transfer the document to the requestor. All received results are valid answers to the queries, and hence there is no need for applying a join algorithm. Although it supports storing large files, they are not stored on the DHT, and the publish/subscribe trees present a maintenance challenge, especially under churn.

b) *Direct storage* - This does not employ any indirection levels, and the keywords are attached to the stored document. The keywords are indexed with a DHT and stored at every responsible node, along with all documents. Though this storage easily supports

	Local storage	**Direct storage**	**Indirect through referencing**
Keyword storage	In the DHT and local storage	Every keyword at every responsible peer in the DHT	Every keyword at its responsible peer in the DHT
Attached to keywords	Pub/Sub tree for every keyword	Document	ObjectID referencing the DHT
Document storage	Local	In the DHT on every responsible peer	in DHT on any other peer

Table 5.1: P2P storage strategies to support keyword-based queries [18]

multiple keyword queries as all the relevant keywords associated with a document are stored together, it results in poor usage of the available limited storage space.

c) *Indirect storage through referencing* - All the keywords are not stored at all their responsible peers but distributed through the DHT to only a single responsible peer. Therefore, the documents are decoupled from the keywords by a single indirection step, and keywords store a short document identifier. Thus there is no need to transfer and store entire documents within the query engine, which necessitates using a join algorithm to evaluate a query with multiple keywords. All the documents that satisfy a single keyword are collected and joined to get the required overall document set.

From these three storage techniques, we select indirect storage through referencing as it supports persistent query answers, persistent document storage, independent document storage, storage of large files, efficient use of storage space, ease of maintenance (update and delete), and has low churn maintenance [18]. However, storage indirection requires the use of join algorithms, especially for multi-keyword queries. The use of storage indirection allows us to utilize the distributed nature of the P2P network, without interfering with the overlay's structure, to perform storage of documents and run keyword-based queries by using the document metadata. Next, we discuss searching for the stored documents using the keywords. After that, we look at how the retrieved documents are filtered out using join techniques to get the required document or documents that fulfill the search criteria.

Search process and searching techniques in DDSs

We now consider the case of storage of documents in a complex data structure such as a distributed data structures (DDSs) like sets, linked-lists and trees [2]. As an example, consider the binary tree illustrated in Figure 5.1. The search query on the binary tree, and by extension other DDSs, proceeds in three phases as follows:

i) *Initialization*: the requesting node R sends the keyword search query to the peer responsible for the keyword, A, which is the root of the binary tree DDS.

ii) *Search*: Node A then sends a broadcast message containing the object ID associated with the keyword and the requestor's peer ID to the nodes that form the DDS to locate the required document.

iii) *Results*: Every matching node in the DDS (C and D) responds directly to the requestor by forwarding the requested document.

To achieve searching in the DDSs, we propose using two techniques: exhaustive and first match search. The *exhaustive search* proceeds through the entire DDS and thoroughly searches

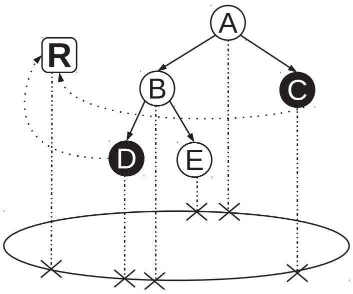
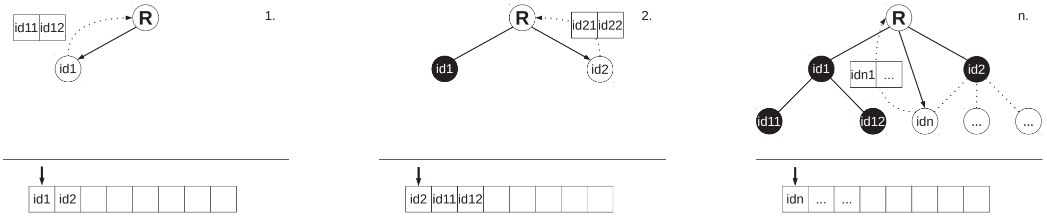


Figure 5.1: DDS structure with corresponding DHT overlay © [2018] IEEE



(a) Step One

(b) Step Two

(c) Step N

Figure 5.2: Simple LocalJoin © [2018] IEEE

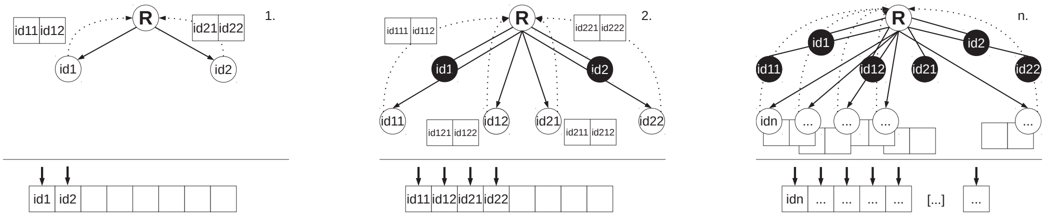
every node. The *first match search* terminates the search once it identifies a match. We do not implement a technique to terminate the broadcast message for the first match search for simplicity during testing and ignore all results after the first match. The required documents are obtained after performing a join involving all the multi-keyword query predicates. Next, we present the join techniques.

Join techniques

The join techniques collate the data received during the query process, especially for multi-keyword queries. In such cases, the query's different predicates will return the associated documents, and the join algorithm then selects the document or documents that fit the entire query. If we consider the mobile phone scenario, then each predicate will return a different set of results, but the required documents must match all three predicates. We propose four algorithms, which we discuss in the following. For ease in illustrating how the algorithms work, we assume a binary tree structure.

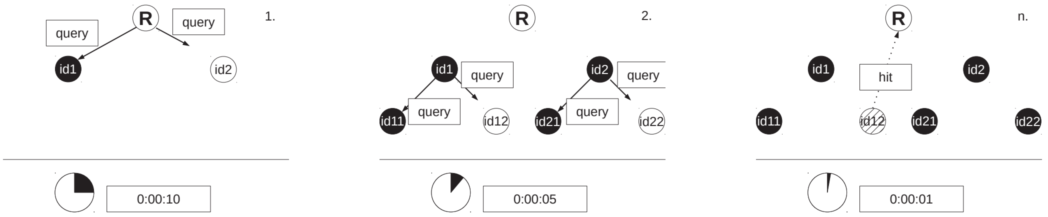
a) *LocalJoin*: The joining process occurs at the requestor. All the data found is sent to the requestor, where the join occurs. This technique is the easiest to implement and has low latency. However, it can cause overloading of the load. It can also generate many messages, for example, when two popular but uncorrelated keywords result in the generation of a large set of documents by their responsible nodes. However, the actual set of documents identified after the local join may be very small. Two LocalJoin techniques are *simple* and *parallel*.

- Simple LocalJoin - The document join process occurs in the order of arrival from the different responding nodes, as shown in Figure 5.2. Theoretically, we expect that



(a) Step One (b) Step Two (c) Step N

Figure 5.3: Parallel LocalJoin © [2018] IEEE



(a) Step One (b) Step Two (c) Step N

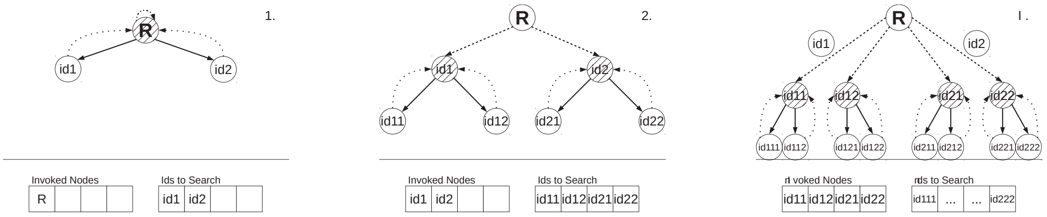
Figure 5.4: Asynchronous NetworkJoin © [2018] IEEE

this algorithm's runtime will grow linearly, but the used data structure can result in different behavior. For example, a stack results in a Depth First Search behavior.

- Parallel LocalJoin - This uses the distributed nature of the network to improve on the simple LocalJoin. Unvisited nodes are contacted in parallel during a single step. Once all results in that round are received, a new set of unvisited nodes is selected. Figure 5.3 shows this process.

b) *Asynchronous NetworkJoin*: In this join algorithm, the requestor node delegates the process of performing the join to the node responsible for the keyword, which in turn delegates it to its child nodes, and so on until a hit is made. Suppose there are further predicates to be met in the query. In that case, the identified document or documents are then forwarded to the next node responsible for the following predicate's keyword, which then delegates the process to its child nodes. This process continues until all predicates are fulfilled, and the final node then forwards the requested documents to the requestor. A timer is incorporated to prevent lengthy delays. Figure 5.4 shows this process. Although NetworkJoin is viewed as having reduced network traffic, an identified negative aspect is the high serial hop count it generates, which results in increased latency.

c) *BloomJoin*: This method utilizes remote method invocation (RMI), as shown in Figure 5.5. Once the requestor sends the query to the node responsible for the keyword, the responsible node first invokes itself to search for its immediate child nodes. It then remotely invokes them to search their children and then compares their payload to identify their child nodes. It then now directly invokes the child nodes of its child nodes to search their child nodes, and this process continues until a hit is made, which is forwarded to the requestor. BloomJoin is expected to have low network traffic, but it has more hop counts than the NetworkJoin hence more latency is anticipated.



(a) Step One

(b) Step Two

(c) Step N

Figure 5.5: BloomJoin © [2018] IEEE

Having highlighted the main design challenges and presented our proposals to address these challenges, we now focus on the experimental setup for testing the proposed solutions.

5.1.2 Experimental setup and testing

To evaluate the quality and the costs incurred by the search mechanism, the metrics we choose are: bandwidth utilized (network traffic), query operation time (time to complete the search), hop counts (number of times a system message is forwarded), reliability ratio (successful operations against started operations), number of messages during the search phase, and average number of nodes contacted. The testbed for the experiment is PeerfactSim.KOM [42, 50, 87], an event-based simulator for P2P networks. The DHT we simulate is based on Chord [139], with replication service supported by an implementation of PAST [35]. We apply the global network position (GNP) model as the networking model, and the churn model is based on measurements in KAD [138]. For testing, we implement three types of DDSs, *binary tree*, *deep tree*, and *customized broad tree*, as shown in Figure 5.6, which we highlight.

i) *Binary tree* - Every node in the tree except for the leaf nodes had two child nodes, resulting in a balanced binary tree with seven levels hence $2^7 = 1024$ nodes. This is shown in Figure 5.6a.

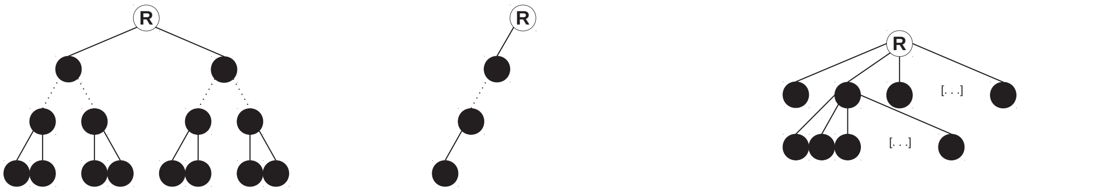
ii) *Deep tree* - The deep tree structure, portrayed in Figure 5.6b, also had 1024 entries. However, unlike the binary tree, the tree nodes, except for the leaf nodes, had only one child node. Therefore, the result is that the number of deep tree levels is equal to the total number of nodes (1024 levels).

iii) *Customized broad tree* - This tree, shown in Figure 5.6c, is designed to be only two-levels deep, with the root node having ten child nodes and each node in the first level having 100 child nodes. Thus the total number of nodes in the network is $1 + 10 + (10 * 100) = 1011$ nodes. It is shown in Figure 5.6c.

5.1.3 Summary of findings

We present the result by considering each of the join algorithms separately.

Simple LocalJoin With the binary tree, this technique has the lowest network traffic, shortest query operation time, least messages generated, and a low number of nodes contacted while achieving the highest reliability ratio. We also see this with the customized tree structure.



(a) Binary tree (b) Deep tree (c) Customized broad tree

Figure 5.6: Tree Structure used in Experiments © [2018] IEEE

However, the simple LocalJoin contacts the most nodes before completing the join. With the deep tree structure, this join algorithm has a lower performance compared with the other algorithms. Considering all metrics, the Simple LocalJoin performs better with the binary tree than the other two DDSs. As is expected, the algorithm incurs lower costs with the first match search than with the exhaustive search.

Parallel LocalJoin Compared to the Simple LocalJoin, this technique generates higher traffic, has a longer query response time, and generates more messages for both the binary tree and customized broad tree. However, it contacts fewer nodes. This comparison holds for both exhaustive and first match searches. For the deep tree, the Parallel LocalJoin performs slightly better than the Simple LocalJoin with exhaustive searching but lower with first match searching. Like the Simple LocalJoin, the performance is better with the binary tree than with the customized broad tree and the deep tree.

Asynchronous NetworkJoin This join technique has the worst performance for the binary tree and the customized broad tree. However, except for messages generated and nodes contacted with the deep tree, its performance would be considered the best. Because the NetworkJoin algorithm only receives the final response after processing by all relevant nodes to fulfill the predicates, it is easy to see why the number of messages generated and the nodes contacted is very high.

BloomJoin The performance of this join technique closely compares with the Simple LocalJoin for the binary tree and the deep tree. With the customized broad tree, it generates twice as much network traffic, requires almost ten times as much time to respond to the query, and generates three times as much messages, while the reliability ratio is much lower. Like both LocalJoin techniques, the BloomJoin is seen to have optimum performance with the binary tree.

Although from the metrics chosen, it appears that the Simple LocalJoin has the best performance when comparing metrics for the binary tree and the customized broad tree, our testing did not consider the impact it has on the local node. The LocalJoin techniques (Simple and Parallel) can overload the peer as all results are sent to a single peer for final processing. Therefore, based on network statistics, the LocalJoin techniques may appear faster and have low network costs. However, based on local node statistics, it may be resource-intensive in terms of processing and storing documents. Therefore the LocalJoin techniques are not as reliable as the network scales upwards. On the other hand, the asynchronous NetworkJoin, although it is network resource-intensive, presents less of a burden to the requestor's local resources. The BloomJoin technique is not suited for networks that utilize, DDSs, such as the customized broad tree, leading to high network resource utilization. Also, remote method invocation is a security challenge, slows down the query processing compared to a local invocation, and presents complexities in implementation.

5.2 Contribution

Our contribution in this paper is the proposal of a metadata-based solution for searching data in a structured P2P network that implements DDSs, such as linked-lists and sets, for complex data storage. We demonstrate the use of storage indirection with referencing in the DDSs to store documents in the network. We also present two techniques to perform searching for the requested documents: exhaustive and first match search. We also discuss four join strategies: simple LocalJoin, parallel LocalJoin, Asynchronous NetworkJoin, and BloomJoin, which we use to filter and consolidate the retrieved data to match the search criteria. We evaluated the search and join approaches using three distributed data trees, binary tree, deep tree, and customized broad tree, to get the performance and costs associated with each join approach. From our results, we deduce that the choice of a join technique depends on the type of DDS and costs expected at the local and network level. The proposed solutions for handling metadata searches in DDSs may prove useful in P2P applications such as online social networks (OSNs) that may need to handle complex data such as albums, comments, and wall posts.

5.3 Personal Contribution

The contributions of the authors are as follows. Raed Al-Aaridhi, Newton Masinde, and Kalman Graffi conceptualized this work. We would also like to acknowledge Daniel Breitlauchs's indirect contribution through his work on the join algorithms done in 2013 as part of his Master's thesis at Paderborn University supervised by Kalman Graffi. Raed Al-Aaridhi, Iakov Dlikman, and Kalman Graffi developed the methodology. Iakov Dlikman carried out the implementation of the solution and the simulations. Computational support and infrastructure needed for the setup of the testing environment were provided by the "Centre for Information and Media Technology" (ZIM) at the Heinrich Heine University (Germany). Raed Al-Aaridhi, Iakov Dlikman, Newton Masinde, and Kalman Graffi performed the results analysis and evaluation, and wrote, reviewed, and edited the paper. In terms of guidance and instruction, the supervision of the work was by Raed Al-Aaridhi and Kalman Graffi.

5.4 Importance and Impact on Thesis

This paper addresses the research question RQ3.1, which focuses on the need for mechanisms to handle the search and retrieval of complex data structures such as albums and comments in P2P-based OSNs using the metadata information. We present two ways of performing the search process and four methods to join the retrieved data to complete the results required by the data requestor's results based on the query issued.

Chapter 6

Caching Structures for Distributed Data Management in P2P-based Social Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [99].

Newton Masinde, Moritz Kanzler, and Kalman Graffi. "Caching Structures for Distributed Data Management in P2P-based Social Networks". In: *Proceedings of the 7th International Symposium on Networks, Computers and Communications (ISNCC 2020)*. IEEE, Oct. 2020, pp. 1–8. DOI: 10.1109/ISNCC49221.2020.9297202
(Received best paper award for Ph.D. students)
Acceptance Rate 45%

In this chapter, we present the second of three proposed mechanisms to improve service delivery, a social caching mechanism for improved data management. The chapter proceeds as follows. In Section 6.1, we present a summary of the paper. We then give the contributions and personal contributions in Section 6.2 and Section 6.3, respectively. Finally, we discuss the importance of the paper and its impact on the thesis in Section 6.4.

6.1 Paper Summary

In this paper, we focus on the possibility of improving social data management in an online social network (OSN). We have seen the concerns raised due to the growth of online social networks (OSNs), that is, accumulated costs of centralization and security & privacy concerns (Chapter 2.1.1). One of the challenges that constitute accumulated costs of centralization has to do with user-generated content management. Due to the social nature of OSNs, the issue that arises is the management of social content or data in a distributed environment [62], which is the problem of scalable dissemination of social updates [67]. Social data/updates refer to data exchanged between users, such as profile information, relationships, community memberships, and statuses. In a centralized OSN, content distribution networks (CDNs) manage this. In decentralized online social networks (DOSNs), the most common solution is social replication/caching of data.

In general, three categories of DOSNs have been identified based on how they undertake data management [62], which are distributed hash table (DHT)-based, social overlay (SO) based, and external resource-based DOSNs. In *DHT-based DOSNs*, the overlays rely on a DHT that stores the social content and offers indexing services. Examples of DOSNs in this class include

DECENT [73], Cachet [113], and LibreSocial [55]. On the other hand, for *social overlay (SO) based DOSNs*, a logical connection node pairs representing existing friendship relations called the social overlay is formed.However, they use the DHT for indexing purposes. An example is DiDuSoNet [61]. Finally *external resource-based DOSNs* may utilize a federation of private web servers such as Diaspora[1] or Vis-à-Vis [135], or rely on cloud-based storage services as Cadros [46]. Therefore these DOSNs are a compromise between fully decentralized and centralized solutions.

6.1.1 Data management in DOSNs

Choosing an appropriate method to handle the data management in DOSNs entails considering data availability/persistence, information diffusion, and privacy [62]. In this paper, we focus on the first two aspects, which we briefly highlight.

a) *Data availability*: Focuses on ensuring the data persists within the distributed environment. Appropriate strategies can be either *external resource-based*, such as using cloud-based services or web servers, or *replica selection*. The replica selection strategies can be either in the DHT or on trusted nodes. For *replication in the DHT* (untrusted nodes), the DHT overlay handles content replication, while for *replication on trusted nodes*, the social network applies measurements to manage the replica placement.

b) *Information diffusion*: Entails considerations into possible techniques for efficient update dissemination and access to needed information/data. Three strategies are distinguishable, request-reply, active dissemination, and hybrid techniques. In *request-reply approaches*, the data is requested for when it is needed, assuming that it availability is guaranteed. This approach is used in DECENT [73], Vegas [38], and LibreSocial [55]. *Active dissemination* performs network flooding with limitations imposed using gossip protocols, such as rumor-mongering or anti-entropy [104], or measures such as Weighted Ego Betweenness Centrality (WEBC) [27] so that only nodes that need such updates receive them. DiDuSoNet [61] is OSN that uses active dissemination. *Hybrid approaches* combine request-reply and active dissemination, such as in Cachet [113].

For improved data availability and reduce the overlay requests active dissemination in DOSNs is preferable. This is easily achievable in SO-based DOSNs as the social overlay controls and stores the social interaction data. However, in DHT-based DOSNs, this presents a challenge as the interactions are under the network overlay's purview and are not visible to the social network. Therefore some ingenuity is needed to introduce support for active dissemination in a DHT-based DOSN. In the next section, we describe a social caching mechanism to solve this challenge.

6.1.2 A DHT-based social caching mechanism

LibreSocial, our peer-to-peer (P2P)-based OSN solution described in [55], is DHT-based. It uses replication on the DHT to guarantee data availability and request-reply for information diffusion. We aim to implement an efficient social caching mechanism that works in tandem with the current caching mechanism and seeks to address the following challenges.

[1] https://diasporafoundation.org

Challenge 1: Use of social updates In DHT-based DOSNs such as LibreSocial, there is no active synchronization between original and cached data. By using social updates, this can be achieved but requires solving two issues.

i) *Detection and distribution of contents* - We propose introducing an instance channel so that content updates are disseminated to a defined receiver list listening to the channel.

ii) *Receiving and storing the updates* - We propose introducing an update channel that listens to instance channels for updates.

Challenge 2: Handling lookup requests We propose integrating a social cache to work within the currently implemented cache's bounds. Thus a data request is serviced by first checking the social cache, then the implemented cache. If the requested data is unavailable, an overlay request then occurs.

Challenge 3: Selecting suitable users based on social interaction We propose two solutions to tackle this problem.

i) *Most-used-contact (MUC) list* - This list contains information about users who make requests from which suitable subscriptions are made. It has a size limitation set to 150 based on the *Dunbar number* [36, 37]. We propose two types of MUC lists, a naïve MUC list that only stores the corresponding user with the number of requests made and an advanced MUC list that stores the corresponding user with the type of request made and the timestamp when it was made.

ii) *Selection algorithms* - We propose three selection algorithms to work in conjunction with the MUC list: *random*, *trend*, and *social score* selection strategies. The random selection strategy uses the naïve MUC list, and as users subscribe to a particular user, they are tracked and added to the list until the limit is reached. Once the limit is attained, unsubscriptions are done randomly to add any newly discovered subscriptions. The trend selection strategy also uses the naïve MUC list and performs a ranking based on the number of lookups of the tracked users in descending order. The n highest users (based on the Dunbar number) to send updates to are then selected. The social score strategy uses the advanced MUC list. It calculates the social score similarly to DiDuSoNet to perform user ranking. The Social score function for LibreSocial $SocialScore_{LS}$ ranks users with higher interaction frequency over a longer time duration as higher than those with fewer interactions over a shorter duration.

Having identified the challenges to be solved and proposed solutions to solve them, we now describe the experimental methodology we apply for testing.

6.1.3 Experimental setup

The quality properties that we are evaluating are *performance*, *local efficiency*, and *overall efficiency*. The metric for performance is the cache hit ratio (cache replies against total replies). The metrics we choose for evaluating local efficiency are social cache size, bootstrap messages, and the average MUC list size per node. Finally, evaluation metrics for overall efficiency are overlay data transferred, average memory consumed per node, and total messages sent. Our test environment constitutes 64 LibreSocial instances running on eight desktop PCs with similar hardware configurations and operating systems (Debian Linux). The Facebook'09 data set [146] is analyzed and down-sampled to get realistic test scenarios based on the user interactions and the time duration. Two test scenario were conducted as follows:

a) Comparison of selection strategies - This is done to identify the most suitable social cache strategy. The Facebook'09 data is down-sampled to 6 hours, and for all instances, the social cache is enabled, and the existing cache mechanism is disabled.

b) Comparing the caching mechanism - The test's duration is two days (48 hours). Four tests are conducted: no-cache, only the current cache enabled (LibreSocial before social caching), only the social caching enabled, and both current and social caching enabled.

6.1.4 Summary of findings

We evaluate the three selection algorithms, random, trend, and social score, against each other by considering their local and overall efficiencies. Although the social score strategy has the lowest cache hit ratio of 92.4% (random-94.9%, trend-93.3%) it is preferred, because it has the smallest overall cache size, fewest overall system messages, and the lowest memory consumption. Finally, we evaluate the caching mechanism's performance for no-cache, the social and current cache each alone, and then combined caches after implementing the social score strategy as the selection method for the social cache. In accordance with our expectations, the social cache displays a tendency for having a larger cache size. A somewhat surprising outcome is that the current cache outperforms the social cache in terms of the cache hit ratio (94.5% against 84.9%). However, when the two caching solutions are used together, we realize a cache hit ratio of 99.2%.

6.2 Contribution

In this paper, we present a social caching mechanism for a DHT-based DOSN, which we implement and evaluate in LibreSocial, our P2P framework for OSNs. Our solution considers the changes that occur in original data and updates the cached data to subscribers actively. By this, we solve one challenge prevalent in DOSNs, that is, the problem of scalable dissemination of social updates. In the social caching mechanism, we solve the challenge of recurrent overlay requests for content to reduce overlay requests and introduce social bootstrapping to address an empty social cache when a new user subscribes. This solution promises to increase data retrievals' efficiency with the combination of the original and social caching, but with considerably more bandwidth usage as the downside due to updates for the social data and current cache refresh requests.

6.3 Personal Contribution

The contributions of the authors are as follows. Newton Masinde, Moritz Kanzler, and Kalman Graffi conceptualized the work. Kalman Graffi provided LibreSocial, the P2P framework for OSNs in its original form, to implement and test our proposed solutions. Newton Masinde, Moritz Kanzler, and Kalman Graffi developed the experimental methodology. Moritz Kanzler implemented and tested the caching solution. Newton Masinde, Moritz Kanzler, and Kalman Graffi analyzed the results. Newton Masinde wrote the introduction, literature, part of the result discussion, and made the paper's conclusions. Moritz Kanzler wrote the sections on the experimental setup and also part of the result discussion. Newton Masinde, Moritz Kanzler, and Kalman Graffi edited and reviewed the final paper. The supervision of the work was done by Newton Masinde and Kalman Graffi.

6.4 Importance and Impact on Thesis

In this paper, we respond to the research question RQ3.2. By designing and implementing a social caching mechanism, we introduce the active dissemination of data to friend nodes in a DHT-based OSN. We show that it is possible to realize reduced overlay requests, hence reduce overall network traffic by introducing the social cache mechanism to work in conjunction with the current caching mechanism. Consequently, more local resources are used to store previously accessed network data, with fewer overlay requests made. This improvement renders the system more scalable as there is less wastage of the often scarce network resources, especially the bandwidth.

Chapter 7

Capacity Management Protocol for a Structured P2P-based Online Social Network

This chapter summarizes the contributions and gives a verbatim copy of our paper [97].

Newton Masinde, Sebastian Bischoff, and Kalman Graffi. "Capacity Management Protocol for a Structured P2P-based Online Social Network". In: *Proceedings of The 7th IEEE Int Conference Social Network Analysis, Management and Security.* SNAMS 2020. IEEE, Dec. 2020, pp. 1–8. ISBN: 978-1-7281-7216-3

In this chapter, we present the last of three proposed mechanisms to improve service delivery, a capacity management protocol to support heterogeneous nodes in a peer-to-peer-based online social network. The chapter proceeds as follows. We present a summary of the paper in Section 7.1. In Section 7.2, we give the contributions, followed by the personal contributions in Section 7.3. Lastly, we highlight the importance and impact of this work on the thesis in Section 7.4.

7.1 Paper Summary

The Internet is a highly heterogeneous network, and this aspect also affects the devices connecting in online social networks (OSNs), with a large percentage of the users connecting via mobile devices. Mobile devices usually have limitations in bandwidth, storage capacity, memory capacities, and many other software-related limitations not present in standard personal computers (PCs). In peer-to-peer (P2P) OSNs, and in particular, distributed hash table (DHT)-based OSNs, handling the presence of heterogeneous devices is non-trivial and requires careful consideration in the development to guarantee practicality and scalability [48]. A good design should incorporate precise models to capture the peer heterogeneity [154] and preferably be abstracted at the lower level while allowing the application layer to perform well. We term the devices with limited capacities as "weak" nodes (or devices). Otherwise, they are "strong" nodes.

The weak nodes do not have sufficient resource capacities (such as storage, processing speed, and bandwidth), and strong nodes have adequate resources to share with the network. We identify two main categories of nodes (user devices):

A - personal computers, notebooks, and similar devices, and

B - mobile devices, such as smartphones and tablets.

We also distinguish the nodes based on how they connect to the network as follows.

I - LAN/WLAN connections, and

II - metered connections often used in mobile communications.

Considering the device and connection types, we identify two large groups, *i*) mostly A-I and B-II, and *ii*) rarely A-II and B-I. Our research primarily focuses on the first group. The strong nodes are in the group A-I and the weak nodes in B-II. The weak nodes should not participate in routing and should not store any replication data, while the strong nodes should support these functions. Meeting this requirement is the challenge in introducing heterogeneity support in P2P-based OSNs, which we present next.

7.1.1 The challenge of heterogeneity management

The basis for the majority of the proposed P2P OSNs are structured overlays, such as Pastry [129] and Chord [139], which are DHT-based. DHTs offer efficient locating of files based on key-based routing (KBR), which promises high reliability and scalability, as well as low deployment costs [90]. However, there is an implicit assumption of uniform distribution of stored objects based on the hash function to guarantee peer load balancing. We also note the emerging problem of uneven file requests (numbers and file sizes), leading to varying storage load and bandwidth consumption. Thus, two challenges arise when DHTs are used in the design of applications [150].

a) *Varying object loads* - The DHT algorithms usually have a load balancing technique, but they do not specify what a load-balanced network means. Further, it does not address the cases of varying file sizes or popular files leading to request overload at some nodes. The load balancing techniques fall into three groups [41].

- Object placement - This ensures a balanced identifier space to prevent load imbalances due to requests. Techniques used include namespace balancing and the use of caching and replication methods.
- Routing - This addresses servicing of lookup queries. The routing algorithm makes use of the routing table. Techniques mostly utilized are link reorganization or path redundancy.
- Underlay load balancing - It is not a guarantee that traffic optimization in the underlying physical network is achievable due to the absence of perfect matching between the nodes and links. Therefore, network proximity techniques, such as topology-based `IDs`, proximity neighbor selection, or proximity routing, are needed.

b) *Varying node capacities* - This requires paying attention to the node's capability (processing power, bandwidth, storage space, and memory) and availability.

To a great degree, the first challenge, handling varying object load, is relatively well solved in many of the DHT-based OSNs. However, node capacity management remains a challenge in many proposed solutions. By addressing this challenge, we envision a DHT-based OSN that, in addition to load balancing, can distinguish the nodes as *weak* and *strong* nodes. We now discuss how LibreSocial addresses the challenge of load and capacity management.

7.1.2 Capacity management in LibreSocial

LibreSocial utilizes a heavily modified version of FreePastry[1], an open-source implementation of Pastry, which is a DHT-based structured overlay. It incorporates the following techniques to handle load management.

a) It achieves a balanced namespace by applying consistent key mapping [66] to guarantee load balancing and uses caching and replication for request load balancing.

b) It utilizes link reorganization which replaces a redundant link with another existing one in a greedy fashion.

c) It implements proximity neighbor selection, which allows for multiple eligible nodes. The closest node based on some metric, such as hop count or actual geographical distance, is selected to fill the routing entry.

The challenge that now presents itself is implementing an efficient protocol for capacity management without adversely affecting the load management feature. For this, we implement an algorithm for capacity management that first identifies the type of device the user is currently using to access the application to eventually classify the node as "strong" or "weak." Even after the initial node classification, the algorithm continually checks the node capacities based on predefined threshold values for memory and storage capacity, type of network connection (wireless or wired), connection bandwidth, battery capacity (for a mobile device), and processing power. Thus, an initially weak node can become a strong node and vice versa. The algorithm incorporates the following restrictions, which make use of the load management feature.

- *Replication* - Only strong nodes are utilized for replica placement. In case a weak node cannot find enough strong nodes for replication, then it is momentarily classified as a strong node. Thus, there is a reduction in the network storage requirement on the weak nodes unless deemed necessary.

- *Routing* - Nodes identified as weak do not maintain routing tables and therefore do not directly participate in the routing of data requests and responses. This change minimizes network traffic via the weak nodes. The weak nodes can still send and receive messages directly to other nodes using the information stored in the leafset.

7.1.3 Experimental setup

The test environment is the high-performance computing (HPC) cluster provided by the "Centre for Information and Media Technology" (ZIM) at Heinrich Heine University. The quality property under investigation is network and storage performance. Our evaluation metrics are average local objects stored, average replicas stored, data storage and retrieval times, and the average lookup times. The test network consists of 200 LibreSocial instances. We design three experiments for which the number of strong nodes is varied to be 1/2, 1/3, and 1/4 of the total network nodes, with a similar workload applied to each experiment. The duration of the experiment is 320 minutes.

[1] https://www.freepastry.org

7.1.4 Summary of findings

The focus of the test was to analyze the storage and network characteristics. We observe that the network data rate increases slightly throughout the experimental duration but does not exceed 20 KB/sec. Hence it is relatively stable. The network message rate for the $^1/_2$ test was on average less than 5000 messages/s, but it was about 40000 messages/sec for the other two experiments. We also note a sharp increase in the average lookup requests handled for the $^1/_4$ experiment (90000 for strong nodes and 350000 for weak nodes), while remaining relatively small for the other two experiments (about 20000 for both weak and strong nodes). Additionally, the average lookup time is generally less than 2 ms, with the spikes occurring at network startup only but never exceeding 16 ms, which we observe for the $^1/_4$ experiment. The routing table for the strong nodes remains fairly stable. However, we see that in cases where failure in the nodes occur, and mainly because weak nodes fail, as is the case with the $^1/_4$ experiment, there was evidence of link reorganization as the routing table sizes changed. When analyzing the storage, we note that strong nodes hold four to five times as many replicas as the weak nodes on average. This result is anticipated and shows that the capacity management protocol differentiates between weak and strong nodes. The average memory contribution, however, slightly increases with an increase in the weak nodes. We suspect that the increase may be due to a significant rise in memory contribution by the strong nodes and not the weak nodes. Another significant observation is the increase in maximum retrieval and storage times, reaching 180 seconds and 1000 seconds, respectively, for the weak nodes for the $^1/_4$ test. This behavior shows that the time it takes to service storage and retrieval requests increases significantly with the increase in the ratio of weak nodes in the network as fewer strong nodes manage the overlay's workload level. Thus it requires more time for a strong node to process tasks. We make the general observation that the network remains functional and stable even when faced with an increased number of weak nodes. However, as the strong nodes decrease below $^1/_2$ of the total nodes, churn indicates that some weak nodes fail.

7.2 Contribution

In this paper, we present a capacity management protocol so as to enhance service delivery in LibreSocial. The inclusion of this protocol in LibreSocial means that it is now capable, to a certain degree, of supporting device heterogeneity. The protocol considers the node capacities to differentiate between nodes that can participate in routing and store replicated data. The algorithm identifies nodes with lower capacities as "weak," else "strong." This classification prevents the weak nodes from being overloaded by the data storage and retrieval requests while also not directly participating in the routing process. Advantageous with the capacity management protocol is that it does not violate the load balancing feature already incorporated in LibreSocial. The results show that even with a large percentage of weak nodes, the network is stable and functional, and remains robust under adverse changes.

7.3 Personal Contribution

The contributions of the paper authors are as follows. Newton Masinde and Kalman Graffi conceptualized the idea for the work. Newton Masinde, Sebastian Bischoff, Kalman Graffi developed the methodology for the research. Newton Masinde and Sebastian Bischoff designed and implemented the capacity management protocol. Newton Masinde performed experimental testing of the protocol. Computational support and infrastructure needed for the setup of the testing environment were provided by the "Centre for Information and Media Technology" (ZIM) at the Heinrich Heine University (Germany). Newton Masinde and Kalman Graffi undertook the result analysis and wrote, reviewed, and edited the paper. Newton Masinde and Kalman Graffi supervised the work.

7.4 Importance and Impact on Thesis

In this paper, the goal is to address the research question RQ3.3. We present a capacity management protocol that is cognizant of the type of device the user connects to the network and the device's hardware capacities to designate the node's status as weak or strong. Hence, strong nodes are considered suitable for handling routing requests and storing replicated data, ensuring that the network, to some degree, supports heterogeneity.

Chapter 8

Conclusion and Future Work

In this chapter, we present a summary of the thesis's contents, followed by a general outlook on the possible future research work. We then present a few closing words to conclude the thesis.

8.1 Conclusion

With the development of the Social Web, the increased popularity and use of online social networks and their influence are evident as individuals and corporations regard them as an integral part of online interactions. OSNs have presented many advantages such as easy and rapid dissemination of information such as news-related items, provide a ready market for advertising of products and services, and offer a platform for users to interact with old friends and make new ones. The development of these OSNs has been under continuous scrutiny. The main concerns raised against the popular OSNs are increased cost overhead due to system scaling and the more insidious security and privacy concerns.

In [98], we primarily identify these concerns as directly related to the fact that the OSN providers run centrally-managed systems. We show that the most viable proposal to solve these concerns is to use decentralized solutions for the OSNs to shift the control from a single provider to the users. In particular, moving to peer-to-peer based OSNs is preferred as it transfers the infrastructural costs to its users and ensures that the users control their data. Further, we derive the technical requirements for designing a P2P framework for OSNs, organizing them into a four-fold software architecture composed of the overlay network, the P2P framework, the application elements, and the graphical user elements. Through a comparative analysis of several proposed P2P-based OSNs, we observe that most solutions implement similar functionalities and strive to solve the security and privacy challenges.

In [55], we present LibreSocial, a P2P framework for OSNs that is a close fit to the proposed software architecture. LibreSocial has been in continuous development since 2008. It is based on a structured P2P overlay, particularly a heavily modified FreePastry, and utilizes the Open Services Gateway Initiative (OSGi™) framework, which supports implementing a modular application design. LibreSocial also includes two plugins, plug-and-play Java ARchive (JAR) files, namely the testing and monitoring plugins, making it possible to perform system testing and collect system statistics for performance evaluation. [100] presents the results of detailed benchmarking tests on LibreSocial. The focus of the testing was on scalability and stability, the ability to handle random user behavior, and the impact of the replication factor on the storage performance. We show that even when the network scales up to 2000 nodes, there is no impairment in the service delivery, and churn does not adversely affect the network stability.

Based on a pseudo-random model to emulate user behavior, we can deduce that the system operates satisfactorily and is deployment-ready. Finally, we observe that the higher the replication factor, the higher the associated costs. However, as the network grows, a low replication factor tends towards poor storage performance.

Having shown that LibreSocial works well, we considered improving the service delivery of the application in general. We presented three improvements: use of metadata to perform data retrieval in distributed data structures, a social caching mechanism for increased data availability, and a capacity management protocol for support of weak and strong nodes.

The first improvement is the metadata-based search mechanisms for data retrieval from the distributed data structures, which we discuss in [1]. We focus on three aspects, the search technique, the join algorithm for retrieved documents, and the type of distributed data structures implemented. We observe that the distributed data structure type and the searching technique played a significant role in selecting the join algorithm as they affect performance and costs at the local and network level in different ways.

For the second improvement, in [99], we introduce a social caching mechanism that utilizes user interaction data (social data) to support the active dissemination of updated data. The complexity of implementing such a solution arises from the fact that DHT-based OSNs do not easily avail the social data as the connections are not social-based. Instead, the interactions are controlled by the overlay. We achieved 99% cache retrieval when the social cache is applied together with the regular cache rather than from the overlay. However, there are considerable system overheads due to updates initiated by the active dissemination algorithm.

The final improvement, presented in [97], introduces heterogeneity support by implementing a capacity management protocol. The protocol identifies the hardware running the application (for example, laptop, desktop, and mobile device) and the type of connection (LAN/WLAN or metered). It then classifies them as weak or strong nodes. This classification aims to prevent weak nodes from participating in the routing and storage of replication data. The protocol actively and continuously checks the nodes to reclassify them by considering their capacities (storage available, bandwidth capacity, memory capacity, and processing power) and type of network connection to check if they fall below a preset threshold value. We show that with even as many as 75% weak nodes, the network remains functional and stable, and the overall performance is not significantly affected.

With these few conclusive statements on the work we present in this thesis, we now focus on possible future contributions based on some of the gaps identified during our work.

8.2 Future Work

LibreSocial presents a rich set of functionalities that we consider to be a step ahead of most proposed P2P-based OSNs. Nevertheless, there is still much room for improving the service delivery and introducing additional functionalities not previously supported to improve usability. While working on this thesis, we interacted with several exciting research topics that we feel can significantly contribute to offering some direction. We discuss them in the following.

8.2.1 Additional OSN functionalities and features

We have identified several application functionalities worth incorporating into LibreSocial and, by extension, any P2P-based OSN. One of these includes an interactive shared calendar. This feature is already available in the centrally managed OSNs. It is generally easier to manage the shared calendar in centralized OSN because the provider handles all the technical aspects, such as calendar synchronization. However, for P2P applications, mechanisms have to be put into place to handle these technical aspects, especially when a user that is to be part of the shared calendar is offline. Another feature to possibly incorporate is a secure multiparty computation module. Secure multiparty computation enables the computation of confidential data, such as statistics on user data, without revealing the input. In the monitoring process, the leaf nodes send the interaction statistics to their parent nodes, which might be considered privacy critical. Using multiparty computation, a set of leaf nodes can compute their monitoring statistics securely. Thus, it allows for the anonymization of the statistics during the monitoring process.

8.2.2 Support for continuous connectivity

Any interactive online application, and especially OSN applications, must necessarily provide mechanisms that ensure continuous connectivity to users within the network. As it stands, guaranteeing continuous connectivity remains a glaring challenge for global P2P applications. For P2P applications, mechanisms that provide an up-to-date list of available bootstrap nodes for joining nodes are necessary. Such a list requires regular maintenance. Also, its provision should be outside of the LibreSocial network, which further introduces a dependency on external services contrary to the philosophy of a fully decentralized OSN. However, if system designers are willing to contend with this, then it can be included in a future iteration of LibreSocial development.

8.2.3 Handling users with criminal intentions

LibreSocial's design puts a priority on ensuring that the network cannot be shut down or censored to support free speech and the flow of ideas. This aspect is possible because its design ensures no administrator and the users have full control and ownership of their data. This aspect of LibreSocial allows for the emergence of another contradictory aspect to LibreSocial's intended use, that is, misuse by criminals. Such users use the network to disburse and share content on issues classified as malicious and against the prevailing ethical standards. In such situations, there is an apparent validity in deleting such content. Therefore, in the future, to solve this emerging challenge, it is proposed to introduce a decentralized, admin-less solution that allows for anonymous reporting and a consensus-based right of a quorum to enforce the deletion of content. However, such solutions may also experience hijacking in a case where most users are malicious, resulting in the removal of good content rather than malicious content. Thus, the solution strongly depends on the fact that most of the consensus users have goodwill. The proposed solution should not hinder resilience against censorship but should simultaneously support democratic mechanisms for moderating illegal content. We believe this introduces the challenge of finding a balance between the system's technical aspects and ethical aspects.

8.3 Closing Words

Research on decentralized solutions for OSNs has been ongoing for almost two decades. P2P-based OSN research was primarily undertaken with the aim of providing OSNs better privacy and security options while allowing the users to retain control of their own data. Therefore, many of the proposed P2P-based OSNs focused on designing and testing novel solutions to handle these aspects of OSNs. However, other technical challenges are apparent by using the P2P platform, such as handling social updates, support for heterogeneous systems, and continuous connectivity. Many of these are already addressed in centralized OSNs. Much of the research focuses on improving service delivery instead of better security and privacy, which is solvable as we have shown with LibreSocial, by assuming a zero-trust network.

We have learned several lessons on the current state of P2P OSN during this work, which we point out in our publications. One very glaring issue with the P2P-based OSN is that there is a relatively small user community compared to the large user base for centralized OSNs. This points to the need for finding ways to motivate an active user base. Also, we observe that many proposals still remain to be research systems which are yet to become active OSNs. This we believe may be partly due to a lack of monetary motivation to push the proposals into fully functional systems. We view monetization as counter-intuitive to the move away from centralized OSNs. The recent trend is that many proposals are hybrids, utilizing some centralized mechanisms such as centralized indexing. Whether purely P2P-based or running a hybrid OSN, the user must be willing to compromise on some features that the application offers.

Acronyms

API application programming interface 5, 22

CDN content distribution network 49

DDS distributed data structure 9–11, 13, 23, 27, 30, 41, 43, 46–48

DHT distributed hash table 6, 7, 11–13, 22, 23, 42, 43, 46, 49, 50, 52, 53, 55–57, 62

DOSN decentralized online social network 10, 13, 17, 18, 49, 50, 52

ECC error-correcting code 23

GUI graphical user interface 10, 21, 29, 33

HTTP Hypertext Transfer Protocol 33

HTTPS Hypertext Transfer Protocol Secure 33

JAR Java ARchive 31, 61

KBR key-based routing 22, 42, 56

MVVM Model-View-View-Model 32

OSGi™ Open Services Gateway Initiative 29, 31, 61

OSN online social network 1–13, 15–27, 29, 30, 33, 35, 39, 40, 48–50, 52, 53, 55, 56, 61, 63

P2P peer-to-peer 5–13, 15, 17, 18, 20–27, 29, 30, 32, 33, 35, 39–43, 46, 48, 50, 52, 55, 56, 63

QoS quality of service 40

RF replication factor 36, 38, 39

RMI remote method invocation 45

UGC user-generated content 3, 16

WEBC Weighted Ego Betweenness Centrality 50

WWW World Wide Web 1

Bibliography

[1] Raed Al-Aaridhi, Iakov Dlikman, Newton Masinde, and Kalman Graffi. “Search Algorithms for Distributed Data Structures in P2P Networks”. In: *Proceedings of the 5th International Symposium on Networks, Computers and Communications (ISNCC 2018)*. IEEE, June 2018, pp. 1–8. DOI: 10.1109/ISNCC.2018.8530977 (Pages: 11, 41, 62, 87).

[2] Raed Al-Aaridhi and Kalman Graffi. “Sets, lists and trees: Distributed data structures on distributed hash tables”. In: *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*. IEEE, Dec. 2016, pp. 1–8. DOI: 10.1109/PCCC.2016.7820639 (Pages: 23, 43).

[3] Lada Adamic and Eytan Adar. “How to search a social network”. In: *Social Networks* 27.3 (2005), pp. 187–203. ISSN: 0378-8733. DOI: 10.1016/j.socnet.2005.01.007 (Page: 2).

[4] Luca Maria Aiello and Giancarlo Ruffo. “LotusNet: Tunable Privacy for Distributed Online Social Network Services”. In: *Computer Communications* 35.1 (Jan. 2012), pp. 75–88. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2010.12.006 (Pages: 3, 4, 19, 20, 25).

[5] Tobias Amft. “The Impact of Resource Sharing on Coexisting P2P Overlays and Stacked Overlay Modules”. PhD thesis. Heinrich-Heine-Universität Düsseldorf, 2017. URL: https://docserv.uni-duesseldorf.de/servlets/DerivateServlet/Derivate-47062/Dissertation-Tobias-Amft-2017-1.pdf (Page: 22).

[6] Tobias Amft and Kalman Graffi. “Moving peers in distributed, location-based peer-to-peer overlays”. In: *2017 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, Jan. 2017, pp. 906–911. DOI: 10.1109/ICCNC.2017.7876253 (Page: 22).

[7] Paul Anderson. *What is Web 2.0? Ideas, Technologies and Implications for Education*. Technical Report: Joint Information Systems Committee (JISC) Bristol, UK, Feb. 2007. URL: http://www.jisc.ac.uk/media/documents/techwatch/tsw0701b.pdf (Page: 1).

[8] Hersh Asthana and Ingemar J. Cox. “PAC’nPOST: A Framework for a Micro-blogging Social Network in an Unstructured P2P Network”. In: *Proceedings of the 21st International Conference on World Wide Web*. WWW ’12 Companion. Lyon, France: ACM, 2012, pp. 455–456. DOI: 10.1145/2187980.2188074 (Page: 24).

[9] Vian Bakir and Andrew McStay. “Fake News and The Economy of Emotions”. In: *Digital Journalism* 6.2 (2018), pp. 154–175. DOI: 10.1080/21670811.2017.1345645 (Pages: 5, 17).

[10] Victor Bekkers, Arthur Edwards, and Dennis de Kool. “Social media monitoring: Responsive governance in the shadow of surveillance?” In: *Government Information Quarterly* 30.4 (2013), pp. 335–342. ISSN: 0740-624X. DOI: 10.1016/j.giq.2013.05.024 (Pages: 5, 17).

[11] Fabrício Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgílio Almeida. "Characterizing User Behavior in Online Social Networks". In: *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. IMC '09. Chicago, Illinois, USA: ACM, 2009, pp. 49–62. ISBN: 9781605587714. DOI: 10.1145/1644893.1644900 (Pages: 12, 37).

[12] Markus Benter, Mohammad Divband, Sebastian Kniesburges, Andreas Koutsopoulos, and Kalman Graffi. "Ca-Re-Chord: A Churn Resistant Self-Stabilizing Chord Overlay Network". In: *2013 Conference on Networked Systems*. IEEE, Mar. 2013, pp. 27–34. DOI: 10.1109/NetSys.2013.11 (Page: 22).

[13] Timothy J. Berners-Lee. *Information management: A proposal*. Technical Report: TBL-900620. CERN, 1989, pp. 1–20 (Page: 1).

[14] Francesco Blasa, Simone Cafiero, Giancarlo Fortino, and Giuseppe Di Fatta. "Symmetric Push-Sum Protocol for Decentralised Aggregation". In: *Proceedings of the Third International Conference on Advances in P2P Systems (AP2PS) 2011)*. Ed. by Antonio Liotta, Nikos Antonopoulus, Guisseppe Di Fatta, Takahiro Hara, and Quang Hieu Vu. International Academy, Research, and Industry Association (IARIA). Lisbon, Portugal, Nov. 2011, pp. 27–32 (Page: 23).

[15] Ewout Bongers and Johan Pouwelse. "A survey of P2P multidimensional indexing structures". In: *arXiv e-prints* (July 2015). Provided by the SAO/NASA Astrophysics Data System. arXiv: 1507.05501 [cs.DC] (Page: 22).

[16] Danah Boyd. "Social network sites: Public, private, or what". In: *Knowledge Tree* 13.1 (2007), pp. 1–7. URL: https://www.danah.org/papers/KnowledgeTree.pdf (Page: 2).

[17] Danah M. Boyd and Nicole B. Ellison. "Social network sites: definition, history, and scholarship". In: *IEEE Engineering Management Review* 38.3 (Mar. 2010), pp. 16–31. ISSN: 1937-4178. DOI: 10.1109/EMR.2010.5559139 (Page: 2).

[18] Daniel Breitlauch. *Search Algorithms for Multi-Criterial Queries in Peer-to-Peer Networks*. Paderborn, Germany, Feb. 2013 (Pages: 42, 43).

[19] Sonja Buchegger, Doris Schiöberg, Le-Hung Vu, and Anwitaman Datta. "PeerSoN: P2P Social Networking: Early Experiences and Insights". In: *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*. SNS '09. Nuremberg, Germany: ACM, 2009, pp. 46–52. ISBN: 978-1-60558-463-8. DOI: 10.1145/1578002.1578010 (Page: 25).

[20] Tomasz Buchert. "Managing large-scale, distributed systems research experiments with control-flows". PhD thesis. Université de Lorraine, Jan. 2016. URL: https://tel.archives-ouvertes.fr/tel-01273964/document (Page: 35).

[21] Min Cai, Martin Frank, Jinbo Chen, and Pedro Szekely. "MAAN: A Multi-Attribute Addressable Network for Grid Information Services". In: *Proceedings of the 4th International Workshop on Grid Computing*. GRID '03. USA: IEEE Computer Society, 2003, pp. 3–4. ISBN: 076952026X. DOI: 10.5555/951948.952051 (Page: 22).

[22] Stéphane Caron, Frédéric Giroire, Dorian Mazauric, Julian Monteiro, and Stéphane Pérennes. "P2P storage systems: Study of different placement policies". In: *Peer-to-Peer Networking and Applications* 7.4 (Dec. 2014), pp. 427–443. ISSN: 1936-6450. DOI: 10.1007/s12083-013-0203-9 (Page: 23).

[23] Manuel Castells. "Virtual Communities or Network Society?" In: *The Internet Galaxy: Reflections on the Internet, Business, and Society*. Oxford University Press, Sept. 2011. Chap. 4, pp. 116–136. DOI: 10.1093/acprof:oso/9780199255771.003.0005 (Page: 2).

[24] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, Animesh Nandi, Antony Rowstron, and Atul Singh. "SplitStream: High-bandwidth Multicast in Cooperative Environments". In: *SIGOPS Oper. Syst. Rev.* 37.5 (Oct. 2003), pp. 298–313. ISSN: 0163-5980. DOI: 10.1145/1165389.945474 (Page: 29).

[25] Karina Clemmons, Amanda Nolen, Judith A. Hayn, and Purnendu Tripathi. "Constructing Community in Higher Education Regardless of Proximity". In: *Handbook of Research on Transnational Higher Education*. Ed. by Siran Mukerji and Purnendu Tripathi. IGI Global, 2014. Chap. 36, pp. 713–729. DOI: 10.4018/978-1-4666-4458-8.ch036 (Page: 2).

[26] Efthymios Constantinides and Stefan J Fountain. "Web 2.0: Conceptual foundations and marketing issues". In: *Journal of Direct, Data and Digital Marketing Practice* 9.3 (2008), pp. 231–244. DOI: 10.1057/palgrave.dddmp.4350098 (Page: 1).

[27] Marco Conti, Andrea De Salve, Barbara Guidi, and Laura Ricci. "Epidemic Diffusion of Social Updates in Dunbar-Based DOSN". In: *Euro-Par 2014: Parallel Processing Workshops*. Cham: Springer, 2014, pp. 311–322. ISBN: 978-3-319-14325-5. DOI: 10.1007/978-3-319-14325-5_27 (Page: 50).

[28] Leucio Antonio Cutillo, Refik Molva, and Thorsten Strufe. "Safebook: A Privacy-Preserving Online Social Network Leveraging on Real-Life Trust". In: *Communications Magazine, IEEE* 47.12 (Dec. 2009), pp. 94–101. DOI: 10.1109/MCOM.2009.5350374 (Page: 25).

[29] Leucio Antonio Cutillo, Refik Molva, and Thorsten Strufe. "Safebook: Feasibility of transitive cooperation for privacy on a decentralized social network". In: *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops*. IEEE, June 2009, pp. 1–6. DOI: 10.1109/WOWMOM.2009.5282446 (Page: 25).

[30] Anwitaman Datta, Sonja Buchegger, Le Hung Vu, Thorsten Strufe, and Krzysztof Rzadca. "Decentralized Online Social Networks". In: *Handbook of Social Network Technologies and Applications*. Ed. by Borko Furht. Boston, MA: Springer, 2010, pp. 349–378. ISBN: 978-1-4419-7142-5. DOI: 10.1007/978-1-4419-7142-5_17 (Page: 17).

[31] Primavera de Filippi and Smari McCarthy. "Cloud computing: Centralization and data sovereignty". In: *European Journal of Law and Technology* 3.2 (2012). URL: https://ssrn.com/abstract=2167372 (Page: 16).

[32] Darcy DiNucci. "Fragmented future". In: *Print* 53.4 (1999). URL: http://darcyd.com/fragmented_future.pdf (Page: 1).

[33] Andreas Disterhöft and Kalman Graffi. "Protected chords in the web: secure P2P framework for decentralized online social networks". In: *2015 IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE, Sept. 2015, pp. 1–5. DOI: 10.1109/P2P.2015.7328520 (Page: 24).

[34] Andreas Disterhöft, Phillip Sandkühler, Andre Ippisch, and Kalman Graffi. "Mr.Tree: Multiple Realities in Tree-based Monitoring Overlays for Peer-to-Peer Networks". In: *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, Mar. 2018, pp. 354–360. DOI: 10.1109/ICCNC.2018.8390361 (Page: 23).

[35] Peter Druschel and Antony Rowstron. "PAST: A large-scale, persistent peer-to-peer storage utility". In: *Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on*. IEEE, May 2001, pp. 75–80. DOI: 10.1109/HOTOS.2001.990064 (Pages: 23, 29, 46).

[36] Robin I. M. Dunbar. “The Social Brain Hypothesis”. In: *Evolutionary Anthropology: Issues, News, and Reviews* 6.5 (1998), pp. 178–190. DOI: 10.1002/(SICI)1520-6505(1998)6:5<178::AID-EVAN5>3.0.CO;2-8 (Page: 51).

[37] Robin I. M. Dunbar. “The Social Brain Hypothesis and its Implications for Social Evolution”. In: *Annals of Human Biology* 36.5 (2009), pp. 562–572. DOI: 10.1080/03014460902960289 (Page: 51).

[38] Michael Dürr, Marco Maier, and Florian Dorfmeister. “Vegas – A Secure and Privacy-Preserving Peer-to-Peer Online Social Network”. In: *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*. IEEE, Sept. 2012, pp. 868–874. DOI: 10.1109/SocialCom-PASSAT.2012.42 (Pages: 25, 50).

[39] Morten Falch, Anders Henten, Reza Tadayoni, and Iwona Windekilde. “Business models in social networking”. In: *CMI International Conference on Social Networking and Communities*. Ballerup, Denmark, 2009. URL: https://vbn.aau.dk/ws/portalfiles/portal/19150157/Falch_3.pdf (Page: 17).

[40] Yuejian Fang, Zilong Wen, Qingni Shen, Yahui Yang, and Zhonghai Wu. “SEDOSN: A Secure Decentralized Online Social Networking Framework”. In: *Embedded System Technology*. Ed. by Xing Zhang, Zhonghai Wu, and Xingmian Sha. Singapore: Springer, 2015, pp. 68–74. ISBN: 978-981-10-0421-6. DOI: 10.1007/978-981-10-0421-6_7 (Page: 25).

[41] Pascal Felber, Peter Kropf, Eryk Schiller, and Sabina Serbu. “Survey on Load Balancing in Peer-to-Peer Distributed Hash Tables”. In: *IEEE Communications Surveys & Tutorials* 16.1 (Jan. 2014), pp. 473–492. ISSN: 2373-745X. DOI: 10.1109/SURV.2013.060313.00157 (Page: 56).

[42] Matthias Feldotto and Kalman Graffi. “Comparative evaluation of peer-to-peer systems using PeerfactSim. KOM”. In: *2013 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE, July 2013, pp. 99–106. DOI: 10.1109/HPCSim.2013.6641399 (Page: 46).

[43] Enrico Franchi, Agostino Poggi, and Michele Tomaiuolo. “Blogracy: A peer-to-peer social network”. In: *International Journal of Distributed Systems and Technologies (IJDST)* 7.2 (2016), pp. 37–56. DOI: 10.4018/IJDST.2016040103 (Page: 25).

[44] Miguel Freitas. “twister - a P2P microblogging platform”. In: *CoRR* abs/1312.7152 (2013), pp. 1–12. arXiv: 1312.7152. URL: https://arxiv.org/pdf/1312.7152.pdf (Page: 25).

[45] Miguel Freitas. “Twister: the development of a peer-to-peer microblogging platform”. In: *International Journal of Parallel, Emergent and Distributed Systems* 31.1 (2016), pp. 20–33. DOI: 10.1080/17445760.2015.1053808 (Page: 25).

[46] Songling Fu, Ligang He, Xiangke Liao, Chenlin Huang, Kenli Li, Cheng Chang, and Bo Gao. “Cadros: The Cloud-Assisted Data Replication in Decentralized Online Social Networks”. In: *Proceedings of the IEEE International Conference on Services Computing*. June 2014, pp. 43–50. DOI: 10.1109/SCC.2014.15 (Page: 50).

[47] Anne J. Gilliland. “Setting the Stage”. In: *Introduction to metadata*. Ed. by Murtha Baca. 3rd ed. Getty Publications, 2016, pp. 1–19. ISBN: 9781606064795. URL: https://www.getty.edu/research/publications/electronic_publications/intrometadata/setting.pdf (Page: 41).

[48] Šarūnas Girdzijauskas, Anwitaman Datta, and Karl Aberer. “Structured Overlay for Heterogeneous Environments: Design and Evaluation of Oscar”. In: *ACM Trans. Auton. Adapt. Syst.* 5.1 (Feb. 2010). ISSN: 1556-4665. DOI: 10.1145/1671948.1671950 (Page: 55).

[49] Jennifer Golbeck and James Hendler. “FilmTrust: movie recommendations using trust in web-based social networks”. In: *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference*. Vol. 1. CCNC 2006. IEEE, Jan. 2006, pp. 282–286. DOI: 10.1109/CCNC.2006.1593032 (Page: 2).

[50] Kalman Graffi. “PeerfactSim. KOM: A P2P system simulator—Experiences and lessons learned”. In: *2011 IEEE International Conference on Peer-to-Peer Computing*. IEEE, Aug. 2011, pp. 154–155. DOI: 10.1109/P2P.2011.6038673 (Page: 46).

[51] Kalman Graffi and Andreas Disterhöft. “SkyEye: A tree-based peer-to-peer monitoring approach”. In: *Pervasive and Mobile Computing* 40 (2017), pp. 593–610. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2017.07.003 (Pages: 11, 23, 31).

[52] Kalman Graffi, Christian Gross, Patrick Mukherjee, Aleksandra Kovacevic, and Ralf Steinmetz. “LifeSocial.KOM: A P2P-based Platform for Secure Online Social Networks”. In: *Proceedings of the IEEE International Conference on Peer-to-Peer Computing (P2P’10)*. IEEE, Aug. 2010, pp. 1–2. DOI: 10.1109/P2P.2010.5569977 (Pages: 24, 29).

[53] Kalman Graffi, Christian Gross, Dominik Stingl, Daniel Hartung, Aleksandra Kovacevic, and Ralf Steinmetz. “LifeSocial. KOM: A Secure and P2P-based Solution for Online Social Networks”. In: *Proceedings of the 2011 IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, Jan. 2011, pp. 554–558. DOI: 10.1109/CCNC.2011.5766541 (Pages: 24, 29).

[54] Kalman Graffi, Aleksandra Kovacevic, Song Xiao, and Ralf Steinmetz. “SkyEye.KOM: An Information Management Over-Overlay for Getting the Oracle View on Structured P2P Systems”. In: *2008 14th IEEE International Conference on Parallel and Distributed Systems*. IEEE, Dec. 2008, pp. 279–286. DOI: 10.1109/ICPADS.2008.8 (Pages: 11, 23, 31).

[55] Kalman Graffi and Newton Masinde. “LibreSocial: A peer-to-peer framework for online social networks”. In: *Concurrency and Computation: Practice and Experience* (Dec. 2020). SPECIAL ISSUE PAPER, pp. 1–26. DOI: 10.1002/cpe.6150 (Pages: 10, 24, 29, 33, 40, 50, 61, 87).

[56] Kalman Graffi, Patrick Mukherjee, Burkhard Menges, Daniel Hartung, Aleksandra Kovacevic, and Ralf Steinmetz. “Practical Security in P2P-Based Social Networks”. In: *Proceedings of the IEEE 34th Conference on Local Computer Networks, 2009. LCN 2009*. IEEE, Oct. 2009, pp. 269–272. DOI: 10.1109/LCN.2009.5355085 (Pages: 24, 29).

[57] Kalman Graffi, Sergey Podrajanski, Patrick Mukherjee, Aleksandra Kovacevic, and Ralf Steinmetz. “A Distributed Platform for Multimedia Communities”. In: *Proceedings of the IEEE International Symposium on Multimedia (ISM’08)*. IEEE, Jan. 2008, pp. 208–213. DOI: 10.1109/ISM.2008.11 (Pages: 7, 24, 29).

[58] Kalman Graffi, Dominik Stingl, Julius Rueckert, Aleksandra Kovacevic, and Ralf Steinmetz. “Monitoring and Management of Structured Peer-to-Peer Systems”. In: *Proceedings P2P 2009, Ninth International Conference on Peer-to-Peer Computing, 9-11 September 2009, Seattle, Washington, USA*. Ed. by Henning Schulzrinne, Karl Aberer, and Anwitaman Datta. IEEE, 2009, pp. 311–320. ISBN: 978-1-4244-5066-4. DOI: 10.1109/P2P.2009.5284512 (Page: 23).

[59] Kálmán György Graffi. "Monitoring and Management of Peer-to-Peer Systems". PhD thesis. Darmstadt University of Technology, 2010. ISBN: 978-3-86853-658-4. URL: https://tuprints.ulb.tu-darmstadt.de/2248/2/Kalman-Graffi_Dissertation_Monitoring-and-Management-of-P2P-Systems.pdf (Page: 23).

[60] Saikat Guha, Kevin Tang, and Paul Francis. "NOYB: Privacy in Online Social Networks". In: *Proceedings of the First Workshop on Online Social Networks*. WOSN '08. Seattle, WA, USA: ACM, 2008, pp. 49–54. ISBN: 9781605581828. DOI: 10.1145/1397735.1397747 (Pages: 2, 17).

[61] Barbara Guidi, Tobias Amft, Andrea De Salve, Kalman Graffi, and Laura Ricci. "DiDuSoNet: A P2P architecture for distributed Dunbar-based social networks". In: *Peer-to-Peer Networking and Applications* 9.6 (2016), pp. 1177–1194. DOI: 10.1007/s12083-015-0366-7 (Pages: 25, 50).

[62] Barbara Guidi, Marco Conti, Andrea Passarella, and Laura Ricci. "Managing social contents in Decentralized Online Social Networks: A Survey". In: *Online Social Networks and Media* 7 (2018), pp. 12–29. ISSN: 2468-6964. DOI: 10.1016/j.osnem.2018.07.001 (Pages: 49, 50).

[63] Barbara Guidi, Marco Conti, and Laura Ricci. "P2P architectures for distributed online social networks". In: *International Conference on High Performance Computing & Simulation, HPCS 2013, Helsinki, Finland, July 1-5, 2013*. IEEE, 2013, pp. 678–681. DOI: 10.1109/HPCSim.2013.6641493 (Pages: 8, 16).

[64] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. "Information Diffusion in Online Social Networks: A Survey". In: *SIGMOD Rec.* 42.2 (July 2013), pp. 17–28. ISSN: 0163-5808. DOI: 10.1145/2503792.2503797 (Page: 2).

[65] Jens Gunstedt, Emmanuel Jeannot, and Martin Quinson. "Experimental Methodologies for Large-Scale Systems: a Survey". In: *Parallel Processing Letters* 19.03 (2009), pp. 399–418. DOI: 10.1142/S0129626409000304 (Page: 35).

[66] Andreas Haeberlen, Jeff Hoye, Alan Mislove, and Peter Druschel. *Consistent key mapping in structured overlays*. Technical Report: TR05-456. CS Department, Rice University, 2005. URL: https://scholarship.rice.edu/bitstream/handle/1911/96342/TR05-456.pdf?sequence=1&isAllowed=y (Page: 57).

[67] Lu Han, Magdalena Punceva, Badri Nath, Shan Muthukrishnan, and Liviu Iftode. "SocialCDN: Caching Techniques for Distributed Social Networks". In: *Proceedings of the IEEE International Conference on Peer-to-Peer Computing (P2P)*. IEEE, Sept. 2012, pp. 191–202. DOI: 10.1109/P2P.2012.6335799 (Page: 49).

[68] Sarah Hartshorn. *5 Differences between social media and social networking*. May 2010. URL: https://www.socialmediatoday.com/content/5-differences-between-social-media-and-social-networking (visited on 06/14/2020) (Page: 2).

[69] Julia Heidemann, Mathias Klier, and Florian Probst. "Online social networks: A survey of a global phenomenon". In: *Computer Networks* 56.18 (2012). The WEB we live in, pp. 3866–3878. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2012.08.009 (Page: 2).

[70] Ai Ho, Abdou Maiga, and Esma Aïmeur. "Privacy protection issues in social networking sites". In: *2009 IEEE/ACS International Conference on Computer Systems and Applications*. IEEE, May 2009, pp. 271–278. DOI: 10.1109/AICCSA.2009.5069336 (Page: 4).

[71] Daniel Lazaro Iglesias, Joan-Manuel Marques, Guillem Cabrera, Helena Rifa-Pous, and Albert Montane. "HorNet: Microblogging for a Contributory Social Network". In: *IEEE Internet Computing* 16.3 (May 2012), pp. 37–45. ISSN: 1089-7801. DOI: 10.1109/MIC.2012.41 (Page: 25).

[72] Jim Isaak and Mina J. Hanna. "User Data Privacy: Facebook, Cambridge Analytica, and Privacy Protection". In: *Computer* 51.8 (Aug. 2018), pp. 56–59. ISSN: 1558-0814. DOI: 10.1109/MC.2018.3191268 (Page: 17).

[73] Sonia Jahid, Shirin Nilizadeh, Prateek Mittal, Nikita Borisov, and Apu Kapadia. "DECENT: A decentralized architecture for enforcing privacy in online social networks". In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*. IEEE, Mar. 2012, pp. 326–332. DOI: 10.1109/PerComW.2012.6197504 (Pages: 24, 50).

[74] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. "T-Man: Gossip-based fast overlay topology construction". In: *Computer Networks* 53.13 (2009). Gossiping in Distributed Systems, pp. 2321–2339. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2009.03.013 (Page: 23).

[75] Badreya Al-Jenaibi. "The nature of Arab public discourse: Social media and the Arab Spring". In: *Journal of Applied Journalism & Media Studies* 3.2 (2014), pp. 241–260. ISSN: 2001-0818. DOI: 10.1386/ajms.3.2.241_1 (Page: 5).

[76] Sam Joseph. "P2P MetaData Search Layers". In: *Agents and Peer-to-Peer Computing*. Ed. by Gianluca Moro, Claudio Sartori, and Munindar P. Singh. Berlin, Heidelberg: Springer, 2005, pp. 101–112. ISBN: 978-3-540-25840-7. DOI: 10.1007/978-3-540-25840-7_11 (Page: 42).

[77] Sam Joseph and Takashige Hoshiai. "Decentralized meta-data strategies: Effective peer-to-peer search". In: *IEICE Transactions on Communications* E86-B.6 (Special Issue on Content Delivery Networks 2003), pp. 1740–1753. ISSN: 0916-8516 (Page: 42).

[78] Pierre St. Juste. "A Peer-to-Peer Architecture for Social Networking Applications". PhD thesis. Gainesvile, Florida: University of Florida, Herbert Wertheim College of Engineering, Department of Electrical and Computer Engineering, 2014. URL: https://ufdcimages.uflib.ufl.edu/UF/E0/04/66/37/00001/ST_JUSTE_P.pdf (Page: 6).

[79] Pierre St. Juste, David Wolinsky, P. Oscar Boykin, and Renato J. Figueiredo. "Litter: A lightweight peer-to-peer microblogging service". In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom)*. IEEE, Oct. 2011, pp. 900–903. DOI: 10.1109/PASSAT/SocialCom.2011.192 (Page: 25).

[80] Imrul Kayes and Adriana Iamnitchi. "Privacy and security in online social networks: A survey". In: *Online Social Networks and Media* 3-4 (2017), pp. 1–21. ISSN: 2468-6964. DOI: 10.1016/j.osnem.2017.09.001 (Pages: 4, 16).

[81] Jan H. Kietzmann, Kristopher Hermkens, Ian P. McCarthy, and Bruno S. Silvestre. "Social media? Get serious! Understanding the functional building blocks of social media". In: *Business Horizons* 54.3 (2011). SPECIAL ISSUE: SOCIAL MEDIA, pp. 241–251. ISSN: 0007-6813. DOI: 10.1016/j.bushor.2011.01.005 (Pages: 1, 18).

[82] Jon Kleinberg. "The Small-World Phenomenon: An Algorithmic Perspective". In: *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*. STOC '00. Portland, Oregon, USA: ACM, 2000, pp. 163–170. ISBN: 1581131844. DOI: 10.1145/335305.335325 (Page: 22).

[83] Dmytri Kleiner. *The Telekommunist Manifesto*. Institute of Network Cultures Amsterdam, 2010. URL: https://www.networkcultures.org/_uploads/%233notebook_telekommunist.pdf (Page: 16).

[84] Neal Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of Computation* 48.177 (1987), pp. 203–209. ISSN: 1088-6842. DOI: 10.1090/S0025-5718-1987-0866109-5 (Page: 30).

[85] Timo Koskela, Otso Kassinen, Erkki Harjula, and Mika Ylianttila. "P2P Group Management Systems: A Conceptual Analysis". In: *ACM Computing Surveys* 45.2 (Mar. 2013), 20:1–20:25. ISSN: 0360-0300. DOI: 10.1145/2431211.2431219 (Page: 22).

[86] Aleksandra Kovacevic, Kalman Graffi, Sebastian Kaune, Christof Leng, and Ralf Steinmetz. "Towards Benchmarking of Structured Peer-to-Peer Overlays for Network Virtual Environments". In: *14th International Conference on Parallel and Distributed Systems, ICPADS 2008, Melbourne, Victoria, Australia, December 8-10, 2008*. IEEE, 2008, pp. 799–804. DOI: 10.1109/ICPADS.2008.68 (Page: 36).

[87] Aleksandra Kovacevic, Sebastian Kaune, Hans Heckel, André Mink, Kalman Graffi, Oliver Heckmann, and Ralf Steinmetz. *PeerfactSim. KOM-A Simulator for Large-Scale Peer-to-Peer Networks*. Technical Report: Tr-2006-06. Technische Universität Darmstadt, Germany, 2006 (Page: 46).

[88] Adam D. I. Kramer, Jamie E. Guillory, and Jeffrey T. Hancock. "Experimental evidence of massive-scale emotional contagion through social networks". In: *Proceedings of the National Academy of Sciences* 111.24 (2014), pp. 8788–8790. ISSN: 0027-8424. DOI: 10.1073/pnas.1320040111 (Pages: 5, 17).

[89] Max Lehn, Christian Triebel Tonio and Gross, Dominik Stingl, Karsten Saller, Wolfgang Effelsberg, Alexandra Kovacevic, and Ralf Steinmetz. "Designing Benchmarks for P2P Systems". In: *From Active Data Management to Event-Based Systems and More: Papers in Honor of Alejandro Buchmann on the Occasion of His 60th Birthday*. Ed. by Kai Sachs, Ilia Petrov, and Pablo Guerrero. Berlin, Heidelberg: Springer, 2010, pp. 209–229. ISBN: 978-3-642-17226-7. DOI: 10.1007/978-3-642-17226-7_13 (Page: 35).

[90] Lichun Li, Xin Xu, Jun Wang, and Wei Wang. "DS2: A DHT-based substrate for distributed services". In: *Peer-to-Peer Networking and Applications* 6.4 (2013), pp. 380–396. DOI: 10.1007/s12083-013-0228-0 (Page: 56).

[91] Nicolas Liebau, Konstantin Pussep, Kalman Graffi, Sebastian Kaune, Andre Beyer, Eric Jahn, and Ralf Steinmetz. "The impact of the P2P paradigm on the new media industries". In: *AMCIS 2007 Proceedings* 255 (2007). URL: https://aisel.aisnet.org/amcis2007/255 (Page: 32).

[92] Andreas Loupasakis, Nikos Ntarmos, and Peter Triantafillou. "eXO: Decentralized Autonomous Scalable Social Networking". In: *CIDR 2011, Fifth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 9-12, 2011, Online Proceedings*. Jan. 2011, pp. 85–95. URL: http://cidrdb.org/cidr2011/Papers/CIDR11_Paper10.pdf (Page: 24).

[93] Apostolos Malatras. "State-of-the-art survey on P2P overlay networks in pervasive computing environments". In: *Journal of Network and Computer Applications* 55 (2015), pp. 1–23. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2015.04.014 (Page: 22).

[94] Tahir Maqsood, Osman Khalid, Rizwana Irfan, Sajjad A. Madani, and Samee U. Khan. "Scalability Issues in Online Social Networks". In: *ACM Computing Surveys* 49.2 (Sept. 2016), 40:1–40:42. ISSN: 0360-0300. DOI: 10.1145/2968216 (Pages: 6, 16, 17).

[95] Newton Masinde. "Implementation of a Sequential Logic Counter". Bachelor's thesis. Nairobi, Kenya: Department of Electrical & Electronic Engineering, Jomo Kenyatta University of Agriculture and Technology, Apr. 2006 (Page: 88).

[96] Newton Masinde. "Robust Weighed Slope One Algorithm". Master's thesis. Hyderabad, India: Department of Computer Science and Engineering, University College of Engineering (Autonomous), Osmania University, Oct. 2013 (Page: 88).

[97] Newton Masinde, Sebastian Bischoff, and Kalman Graffi. "Capacity Management Protocol for a Structured P2P-based Online Social Network". In: *Proceedings of The 7th IEEE Int Conference Social Network Analysis, Management and Security*. SNAMS 2020. IEEE, Dec. 2020, pp. 1–8. ISBN: 978-1-7281-7216-3 (Pages: 11, 55, 62, 87).

[98] Newton Masinde and Kalman Graffi. "Peer-to-Peer based Social Networks: A Comprehensive Survey". In: *SN Computer Science* 1.5 (2020), pp. 1–51. DOI: 10.1007/s42979-020-00315-8 (Pages: 10, 15, 22, 24, 25, 33, 61, 87).

[99] Newton Masinde, Moritz Kanzler, and Kalman Graffi. "Caching Structures for Distributed Data Management in P2P-based Social Networks". In: *Proceedings of the 7th International Symposium on Networks, Computers and Communications (ISNCC 2020)*. IEEE, Oct. 2020, pp. 1–8. DOI: 10.1109/ISNCC49221.2020.9297202 (Pages: 11, 49, 62, 87).

[100] Newton Masinde, Liat Khitman, Iakov Dlikman, and Kalman Graffi. "Systematic Evaluation of LibreSocial–A Peer-to-Peer Framework for Online Social Networks". In: *Future Internet* 12.9 (2020), p. 140. DOI: 10.3390/fi12090140 (Pages: 11, 35, 40, 61, 87).

[101] Newton Masinde, Gerald Schaefer, and Iakov Korovin. "Stable and reliable predictive accuracy of robust weighted slope one under profile injection attacks". In: *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE, May 2016, pp. 1041–1046. DOI: 10.1109/ICIEV.2016.7760157 (Page: 87).

[102] Newton W. Masinde and Sameen S. Fatima. "Effect of varying filler-size in profile injection attacks on the Robust Weighted Slope One". In: *Proceedings of the 2nd Pan African International Conference on Science, Computing and Telecommunications (PACT 2014)*. IEEE, July 2014, pp. 92–97. DOI: 10.1109/SCAT.2014.7055125 (Page: 87).

[103] Petar Maymounkov and David Mazieres. "Kademlia: A peer-to-peer information system based on the xor metric". In: *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers*. Ed. by Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron. Vol. 2429. Lecture Notes in Computer Science. Springer, 2002, pp. 53–65. DOI: 10.1007/3-540-45748-8_5 (Page: 30).

[104] Giuliano Mega, Alberto Montresor, and Gian Pietro Picco. "Efficient Dissemination in Decentralized Social Networks". In: *Proceedings of the IEEE International Conference on Peer-to-Peer Computing*. IEEE, Aug. 2011, pp. 338–347. DOI: 10.1109/P2P.2011.6038753 (Page: 50).

[105] Victor S. Miller. "Use of Elliptic Curves in Cryptography". In: *Advances in Cryptology — CRYPTO '85 Proceedings*. Ed. by Hugh C. Williams. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 417–426. ISBN: 978-3-540-39799-1. DOI: 10.1007/3-540-39799-X_31 (Page: 30).

[106] San Murugesan. "Understanding Web 2.0". In: *IT Professional* 9.4 (2007), pp. 34–41 (Page: 15).

[107] Katarzyna Musiał and Przemysław Kazienko. “Social networks on the internet”. In: *World Wide Web* 16.1 (2013), pp. 31–72. DOI: 10.1007/s11280-011-0155-z (Page: 3).

[108] Rammohan Narendula, Thanasis G. Papaioannou, and Karl Aberer. “My3: A highly-available P2P-based online social network”. In: *Peer-to-Peer Computing (P2P), 2011 IEEE International Conference on*. IEEE, Aug. 2011, pp. 166–167. DOI: 10.1109/P2P.2011.6038730 (Page: 24).

[109] Rammohan Narendula, Thanasis G. Papaioannou, and Karl Aberer. “Privacy-aware and highly-available OSN profiles”. In: *2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*. IEEE, June 2010, pp. 211–216. DOI: 10.1109/WETICE.2010.40 (Page: 24).

[110] National Research Council. *Funding a Revolution: Government Support for Computing Research*. Washington, DC: The National Academy Press, 1999. Chap. 7, pp. 179–180. ISBN: 0-309-06278-0. DOI: 10.17226/6323 (Page: 1).

[111] Surya Nepal, Cécile Paris, and Athman Bouguettaya. “Trusting the social web: issues and challenges”. In: *World Wide Web* 18.1 (2015), pp. 1–7. DOI: 10.1007/s11280-013-0252-2 (Page: 2).

[112] Blaise Ngonmang, Emmanuel Viennet, Savaneary Sean, Philippe Stepniewski, Françoise Fogelman-Soulié, and Rémi Kirche. “Monetization and Services on a Real Online Social Network Using Social Network Analysis”. In: *2013 IEEE 13th International Conference on Data Mining Workshops*. IEEE, Dec. 2013, pp. 185–193. DOI: 10.1109/ICDMW.2013.78 (Page: 16).

[113] Shirin Nilizadeh, Sonia Jahid, Prateek Mittal, Nikita Borisov, and Apu Kapadia. “Cachet: A Decentralized Architecture for Privacy Preserving Social Networking with Caching”. In: *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*. CoNEXT ’12. Nice, France: ACM, 2012, pp. 337–348. ISBN: 978-1-4503-1775-7. DOI: 10.1145/2413176.2413215 (Pages: 25, 50).

[114] Tim O’Reilly. “What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software”. In: *Communications & Strategies* 1 (2007), p. 17 (Pages: 1, 15).

[115] Tim O’Reilly. “What is Web 2.0?: Design Patterns and Business Models for the Next Generation of Software”. In: (2010). Ed. by Helen Donelan, Karen Kear, and Magnus Ramage (Page: 1).

[116] Jonathan A. Obar and Steve Wildman. “Social media definition and the governance challenge: An introduction to the special issue”. In: *Telecommunications Policy* 39.9 (2015). SPECIAL ISSUE ON THE GOVERNANCE OF SOCIAL MEDIA, pp. 745–750. ISSN: 0308-5961. DOI: 10.1016/j.telpol.2015.07.014 (Pages: 1, 2).

[117] Samia Oukemeni, Helena Rifà-Pous, and Joan Manuel Marquès Puig. “Privacy Analysis on Microblogging Online Social Networks: A Survey”. In: *ACM Comput. Surv.* 52.3 (June 2019), 60:1–60:36. ISSN: 0360-0300. DOI: 10.1145/3321481 (Page: 16).

[118] George Pallis, Demetrios Zeinalipour-Yazti, and Marios D. Dikaiakos. “Online Social Networks: Status and Trends”. In: *New Directions in Web Data Management 1*. Ed. by Athena Vakali and Lakhmi C. Jain. Berlin, Heidelberg: Springer, 2011, pp. 213–234. ISBN: 978-3-642-17551-0. DOI: 10.1007/978-3-642-17551-0_8 (Page: 4).

[119] Thomas Paul, Sonja Buchegger, and Thorsten Strufe. "Decentralized Social Networking Services". In: *Trustworthy Internet*. Ed. by Luca Salgarelli, Giuseppe Bianchi, and Nicola Blefari-Melazzi. Milano: Springer, 2011, pp. 187–199. ISBN: 978-88-470-1818-1. DOI: 10.1007/978-88-470-1818-1_14 (Page: 17).

[120] Thomas Paul, Antonino Famulari, and Thorsten Strufe. "A survey on decentralized Online Social Networks". In: *Computer Networks* 75 (2014), pp. 437–452. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2014.10.005 (Pages: 5–7, 17).

[121] Timothy Perfitt and Burkhard Englert. "Megaphone: Fault Tolerant, Scalable, and Trustworthy P2P Microblogging". In: *2010 Fifth International Conference on Internet and Web Applications and Services*. IEEE, May 2010, pp. 469–477. DOI: 10.1109/ICIW.2010.77 (Page: 24).

[122] Bob Quinn and Kevin Almeroth. *IP Multicast Applications: Challenges and Solutions*. RFC 3170. Sept. 2001. DOI: 10.17487/RFC3170. URL: https://www.rfc-editor.org/rfc/rfc3170.txt (Page: 23).

[123] Fatemeh Raji, Mohammad Davarpanah Jazi, and Ali Miri. "PESCA: a peer-to-peer social network architecture with privacy-enabled social communication and data availability". In: *IET Information Security* 9.1 (Jan. 2015), pp. 73–80. DOI: 10.1049/iet-ifs.2013.0256 (Page: 24).

[124] Vitaliy Rapp and Kalman Graffi. "Continuous Gossip-Based Aggregation through Dynamic Information Aging". In: *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, July 2013, pp. 1–7. DOI: 10.1109/ICCCN.2013.6614118 (Page: 23).

[125] Irving S. Reed and Gustave Solomon. "Polynomial Codes Over Certain Finite Fields". In: *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), pp. 300–304. DOI: 10.1137/0108018 (Page: 23).

[126] Alexander Richter and Michael Koch. "Functions of Social Networking Services". In: *Proceedings of the 8th International Conference on Designing Cooperative Systems, COOP 2008, Carry-le-Rouet, Provence, France, May 20-23, 2008*. ACM Press, 2008, pp. 87–98. URL: https://dl.eusset.eu/handle/20.500.12015/2802 (Pages: 3, 4).

[127] John Risson and Tim Moors. "Survey of Research Towards Robust Peer-to-Peer Networks: Search Methods". In: *Computer Networks* 50.17 (2006), pp. 3485–3521. DOI: 10.1016/j.comnet.2006.02.001 (Page: 22).

[128] Suzanne Robertson and James Robertson. *Mastering the Requirements Process: Getting Requirements Right*. 3rd ed. Addison-Wesley, 2012. ISBN: 978-0-321-81574-3 (Pages: 19, 20).

[129] Antony Rowstron and Peter Druschel. "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems". In: *Middleware 2001, IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, Germany, November 12-16, 2001, Proceedings*. Vol. 2218. Lecture Notes in Computer Science. Springer, 2001, pp. 329–350. DOI: 10.1007/3-540-45518-3_18 (Pages: 6, 22, 29, 30, 56).

[130] Antony Rowstron and Peter Druschel. "Storage Management and Caching in PAST, a Large-scale, Persistent Peer-to-peer Storage Utility". In: *SIGOPS Oper. Syst. Rev.* 35.5 (Oct. 2001), pp. 188–201. ISSN: 0163-5980. DOI: 10.1145/502059.502053 (Pages: 23, 29).

[131] Antony Rowstron, Anne-Marie Kermarrec, Peter Druschel, and Miguel Castro. "Scribe: The Design of a Large-Scale Event Notification Infrastructure". In: *Networked Group Communication: Third International COST264 Workshop, NGC 2001 London, UK, November 7–9, 2001 Proceedings*. Ed. by Jon Crowcroft. Springer Berlin Heidelberg, 2001, pp. 30–43. ISBN: 978-3-540-45546-2. DOI: 10.1007/3-540-45546-9_3 (Page: 29).

[132] Rianka Roy and Nilanjana Gupta. "Digital Capitalism and Surveillance on Social Networking Sites: A Study of Digital Labour, Security and Privacy for Social Media Users". In: *Digital India: Reflections and Practice*. Ed. by Arpan Kumar Kar, Shuchi Sinha, and M. P. Gupta. Cham: Springer, 2018, pp. 67–81. ISBN: 978-3-319-78378-9. DOI: 10.1007/978-3-319-78378-9_4 (Pages: 5, 17).

[133] Karsten Saller, Kamill Panitzek, and Max Lehn. "Benchmarking Methodology". In: *Benchmarking Peer-to-Peer Systems: Understanding Quality of Service in Large-Scale Distributed Systems*. Ed. by Wolfgang Effelsberg, Ralf Steinmetz, and Thorsten Strufe. Berlin, Heidelberg: Springer, 2013, pp. 19–45. ISBN: 978-3-642-38673-2. DOI: 10.1007/978-3-642-38673-2_3 (Page: 36).

[134] Peter Schauer. *5 Differences between social media and social networking*. June 2015. URL: https://www.socialmediatoday.com/social-business/peteschauer/2015-06-28/5-biggest-differences-between-social-media-and-social (visited on 06/14/2020) (Page: 2).

[135] Amre Shakimov, Harold Lim, Ramón Cáceres, Landon P. Cox, Kevin A. Li, Dongtao Liu, and Alexander Varshavsky. "Vis-à-Vis: Privacy-preserving Online Social Networking via Virtual Individual Servers". In: *Proceedings of the International Conference on Communication Systems and Networks (COMSNETS'11)*. IEEE, Jan. 2011, pp. 1–10. DOI: 10.1109/COMSNETS.2011.5716497 (Page: 50).

[136] Rajesh Sharma and Anwitaman Datta. "SuperNova: Super-Peers Based Architecture for Decentralized Online Social Networks". In: *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*. IEEE, Jan. 2012, pp. 1–10. DOI: 10.1109/COMSNETS.2012.6151349 (Page: 25).

[137] Sarah Spiekermann, Alessandro Acquisti, Rainer Böhme, and Kai-Lung Hui. "The challenges of personal data markets and privacy". In: *Electronic Markets* 25.2 (June 2015), pp. 161–167. ISSN: 1422-8890. DOI: 10.1007/s12525-015-0191-0 (Pages: 5, 17).

[138] Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. "Long Term Study of Peer Behavior in the kad DHT". In: *IEEE/ACM Transactions on Networking* 17.5 (Oct. 2009), pp. 1371–1384. ISSN: 1558-2566. DOI: 10.1109/TNET.2008.2009053 (Page: 46).

[139] Ion Stoica, Robert Morris, David Karger, Marinus Frans Kaashoek, and Hari Balakrishnan. "Chord: A scalable peer-to-peer lookup service for internet applications". In: *Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 27-31, 2001, San Diego, CA, USA*. ACM, 2001, pp. 149–160. DOI: 10.1145/383059.383071 (Pages: 22, 46, 56).

[140] Elizabeth Stoycheff. "Under Surveillance: Examining Facebook's Spiral of Silence Effects in the Wake of NSA Internet Monitoring". In: *Journalism & Mass Communication Quarterly* 93.2 (2016), pp. 296–311. DOI: 10.1177/1077699016630255 (Pages: 5, 17).

[141] Xiaoping Sun. “SCAN: A Small-World Structured P2p Overlay for Multi-Dimensional Queries”. In: *Proceedings of the 16th International Conference on World Wide Web.* WWW ’07. Banff, Alberta, Canada: ACM, 2007, pp. 1191–1192. ISBN: 9781595936547. DOI: 10.1145/1242572.1242760 (Page: 22).

[142] Sanaz Taheri-Boshrooyeh, Alptekin Küpçü, and Öznur Özkasap. “Security and Privacy of Distributed Online Social Networks”. In: *2015 IEEE 35th International Conference on Distributed Computing Systems Workshops.* IEEE, June 2015, pp. 112–119. DOI: 10.1109/ICDCSW.2015.30 (Page: 18).

[143] Sebastian Tramp, Philipp Frischmuth, Timofey Ermilov, Saeedeh Shekarpour, and Sören Auer. “An architecture of a distributed semantic social network”. In: *Semantic Web* 5.1 (2014), pp. 77–95. DOI: 10.3233/SW-2012-0082 (Page: 17).

[144] Ha Manh Tran, Van Sinh Nguyen, and Synh Viet Uyen Ha. “Decentralized Online Social Network Using Peer-to-Peer Technology”. In: *REV Journal on Electronics and Communications* 5.1-2 (June 2015). DOI: 10.21553/rev-jec.95 (Pages: 24, 25).

[145] Maarten van Steen and Andrew S. Tanenbaum. “A brief introduction to distributed systems”. In: *Computing* 98.10 (2016), pp. 967–1009. DOI: 10.1007/s00607-016-0508-7 (Page: 16).

[146] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. “On the Evolution of User Interaction in Facebook”. In: *Proceedings of the ACM Workshop on Online Social Networks.* WOSN ’09. Barcelona, Spain: ACM, 2009, pp. 37–42. ISBN: 978-1-60558-445-4. DOI: 10.1145/1592665.1592675 (Page: 51).

[147] Jinbao Wang, Sai Wu, Hong Gao, Jianzhong Li, and Beng Chin Ooi. “Indexing Multi-Dimensional Data in a Cloud System”. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data.* SIGMOD ’10. Indianapolis, Indiana, USA: ACM, 2010, pp. 591–602. ISBN: 9781450300322. DOI: 10.1145/1807167.1807232 (Page: 22).

[148] Jiwei Wang, Fangai Liu, Xu Li, Haoran Liu, and Xiaohui Zhao. “HPOSN: A Novel Online Social Network Model Based on Hybrid P2P”. In: *2015 International Conference on Cloud Computing and Big Data (CCBD).* IEEE, Nov. 2015, pp. 342–349. DOI: 10.1109/CCBD.2015.46 (Page: 25).

[149] Barry Wellman, Janet Salaff, Dimitrina Dimitrova, Laura Garton, Milena Gulia, and Caroline Haythornthwaite. “Computer Networks as Social Networks: Collaborative Work, Telework, and Virtual Community”. In: *Annual Review of Sociology* 22.1 (1996), pp. 213–238. DOI: 10.1146/annurev.soc.22.1.213 (Page: 2).

[150] Philip Wette and Kalman Graffi. “Adding Capacity-Aware Storage Indirection to Homogeneous Distributed Hash Tables”. In: *2013 Conference on Networked Systems.* IEEE, Mar. 2013, pp. 35–42. DOI: 10.1109/NetSys.2013.9 (Pages: 30, 56).

[151] Karl Wiegers and Joy Beatty. *Software Requirements.* 3rd ed. Microsoft Press, 2013. ISBN: 978-0-7356-7966-5 (Pages: 19, 20).

[152] Jie Wu. *Distributed System Design.* 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1998. ISBN: 0849331781 (Page: 15).

[153] Tianyin Xu, Yang Chen, Jin Zhao, and Xiaoming Fu. “Cuckoo: towards decentralized, socio-aware online microblogging services and data measurements”. In: *Proceedings of the 2nd ACM International Workshop on Hot Topics in Planet-scale Measurement.* HotPlanet ’10 4. San Francisco, California: ACM, 2010, 4:1–4:6. DOI: 10.1145/1834616.1834622 (Page: 25).

[154] Zhi Yang, Yuanjian Xing, Feng Xiao, Zhi Qu, Xiaoming Li, and Yafei Dai. "Exploring peer heterogeneity: Towards understanding and application". In: *2011 IEEE International Conference on Peer-to-Peer Computing*. IEEE, Aug. 2011, pp. 20–29. DOI: 10.1109/P2P.2011.6038657 (Page: 55).

[155] Ralph Rowland Young. *The Requirements Engineering Handbook*. Artech House, 2004. ISBN: 1-58053-266-7 (Page: 19).

[156] Chi Zhang, Jinyuan Sun, Xiaoyan Zhu, and Yuguang Fang. "Privacy and Security for Online Social Networks: Challenges and Opportunities". In: *Network, IEEE* 24.4 (July 2010), pp. 13–18. ISSN: 1558-156X. DOI: 10.1109/MNET.2010.5510913 (Pages: 3, 4, 19).

[157] Chong Zhang, Weidong Xiao, Daquan Tang, and Jiuyang Tang. "P2P-based multidimensional indexing methods: A survey". In: *Journal of Systems and Software* 84.12 (Dec. 2011), pp. 2348–2362. ISSN: 0164-1212. DOI: 10.1016/j.jss.2011.07.027 (Page: 22).

Newton Wafula Masinde

Personal Information

Date of birth:	6th September, 1981
Place of birth:	Nakuru, Kenya
Nationality:	Kenyan
Marital Status:	Married to Pauline Winnie Orondo
Languages:	English, Swahili, German

Contact data

Address:	Im Melchersfeld 32 41468, Neuss, Germany
Mobile:	+49 1521 701 2099
email:	amphinewt@gmail.com

Education

2017-ToDate Doctoral research in Computer Science
Heinrich Heine University
Universitätsstr. 1, 40225 Düsseldorf, Germany
Website: tsn.hhu.de
Thesis Title: *"Peer-to-Peer Mechanisms for Fully Decentralized, Secure and Scalable Online Social Networks"*
Thesis Supervisor: Jun.-Prof. Dr.-Ing. Kalman Graffi

2011-2013 Master of Technology (M.Tech) in Computer Science and Engineering
University College of Engineering, Osmania University
Telangana State, 500007 Hyderabad, India
Website: www.uceou.edu
Thesis Title: *"Robust Weighted Slope One Algorithm"*
Thesis Supervisor: Prof. Sameen S. Fatima

2001-2006 Bachelor of Science in Electronic and Computer Engineering
Jomo Kenyatta University of Agriculture and Technology
Juja-Main Campus, P.O. Box 62000-00200 Nairobi, Kenya
Website: www.jkuat.ac.ke
Project Title: *"Implementation of a Sequential Logic Counter"*
Project Supervisor: Mr. Savero L. Ogaba

1996-1999 Kenya Certificate of Secondary Education
Sunshine Secondary School
P.O. Box 56890-00100, Nairobi, Kenya
Website: sunshineschool.sc.ke
Final grade: *"B+"* (80 points)

Work Experience

Actual	Heinrich Heine University *Research Assistant*
01/2014-to date	Jaramogi Oginga Odinga Univ. of Science and Technology *Tutorial Fellow*
05/2008 - 08/2010	Pegrume (K) Limited *Assistant Engineer*
01/2008-04/2008	McCann-Erickson Advertising, Kenya *Information Technology Assistant*
10/2007-12/2007	TechnoBrain (K) Ltd *Intern Tutor*
05/2005–06/2005	Kenya Airways Ltd *Information Systems Intern*
05/2004–07/2004	Kenya Airways Ltd *Information Systems Intern*
01/2003–03/2003	NairobiNet Online (K) Ltd *Client Support Intern*

Professional Certifications

11/2013	Red Hat Certified System Administrator (RCSA) Red Hat Enterprise Linux 6 (Certification No. 130-204-670)
08/2009	Microsoft Certified Systems Administrator (MCSA) Messaging on Windows Server 2003 (Certificate No. A719-0661)
08/2009	Microsoft Certified Systems Engineer (MCSE) Windows Server 2003 (Certificate No. A719-0659)
08/2009	Microsoft Certified Systems Administrator (MCSA) Windows Server 2003 (Certificate No. A719-0658)
02/2007	Microsoft Certified Professional (Certificate No. C090-7152)
05/2007	Oracle Database 10g Administrator Certified Associate Version Retired

Scholarships and Awards

2011	Cultural Exchange Program (CEP) Scholarship Indian Council for Cultural Relations (ICCR) Purpose: *Masters studies*
2016	Kenyan - German Postgraduate Training Programme 2016/2017 Scholarship Deutscher Akademischer Austauschdienst (DAAD) Purpose: *Doctoral research*

Memberships to Professional Bodies

Institute of Electrical and Electronic Engineers (IEEE)	Student member Member No. *92176100*

Conferences and Workshops attended

Conferences

1. *The 7th IEEE Int Conference Social Network Analysis, Management and Security* (SNAMS 2020) held in Paris, France on the 14-16 December, 2020. Technically Co-Sponsored by IEEE France Section.

2. *7th edition of the International Symposium on Networks, Computers and Communications* (ISNCC 2020) held in Montreal, Canada on the 20-22 October, 2020. Technically sponsored by IEEE and IEEE Communications Society.

3. *The 2015 Pan African International Conference on Science, Computing and Telecommunications* (PACT 2015) held at Fairway Hotel, 1-2 Kafu Road, Kampala, Uganda on the 27–29 July, 2015. Technically sponsored by IEEE.

4. *The 2014 Pan African International Conference on Science, Computing and Telecommunications* (PACT 2014) held at the Nelson Mandela African Institute of Science and Technology, Arusha, Tanzania on the 14-18 July, 2014. Technically sponsored by IEEE.

Workshops and Forums

5. Virtual Heidelberg Laureate Forum *"Traversing Separation"* - 21-25 September, 2020.

6. *Project Management in Science*: Workshop organized by Interdisciplinary Graduate and Research Academy Düsseldorf (iGRAD) - 27th May and 5th June, 2020.

7. *Get into Teaching for Doctoral Researchers*: Workshop organized by Interdisciplinary Graduate and Research Academy Düsseldorf (iGRAD) on the 7/8 November, 2019.

8. *Presenting (in) Science - How to own the stage on (international) conferences*: Workshop organized by Interdisciplinary Graduate and Research Academy Düsseldorf (iGRAD) on the 15/16 August, 2019.

9. *Good Scientific Practice for Doctoral Researchers*: Workshop organized by Interdisciplinary Graduate and Research Academy Düsseldorf (iGRAD) on the 21st August, 2017.

10. *Pedagogical Skills Workshop for Academic Staff* held at Jaramogi Oginga Odinga University of Science and Technology (JOOUST) on 6th June 2014.

Publications

Theses

1. **Newton Masinde**. “Robust Weighed Slope One Algorithm”. Master thesis. Hyderabad, India: Osmania University, Oct. 2013.

2. **Newton Masinde**. “Implementation of a Sequential Logic Counter”. Bachelor thesis. Nairobi, Kenya: Department of Electronic, Computer Engineering, Jomo Kenyatta University of Agriculture, and Technology, 2006

Articles

3. Kalman Graffi and **Newton Masinde**. “LibreSocial: A peer-to-peer framework for online social networks”. In: *Concurrency and Computation: Practice and Experience* (Dec. 2020). SPECIAL ISSUE PAPER, pp. 1–26. doi: 10.1002/cpe.6150

4. **Newton Masinde** and Kalman Graffi. “Peer-to-Peer based Social Networks: A Comprehensive Survey”. In: *SN Computer Science* 1.5 (2020), pp. 1–51. doi: 10.1007/s42979-020-00315-8

5. **Newton Masinde**, Liat Khitman, Iakov Dlikman, and Kalman Graffi. “Systematic Evaluation of LibreSocial—A Peer-to-Peer Framework for Online Social Networks”. In: *Future Internet* 12.140 (9 2020). doi: 10.3390/fi12090140

Conference Proceedings

6. **Newton Masinde**, Sebastian Bischoff, and Kalman Graffi. “Capacity Management Protocol for a Structured P2P-based Online Social Network”. In: *Proceedings of The 7th IEEE Int Conference Social Network Analysis, Management and Security.* SNAMS 2020. IEEE, Dec. 2020, pp. 1–8. isbn: 978-1-7281-7216-3

7. **Newton Masinde**, Moritz Kanzler, and Kalman Graffi. “Caching Structures for Distributed Data Management in P2P-based Social Networks”. In: *Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC).* IEEE, Oct. 2018

8. Raed Al-Aaridhi, Iakov Dlikman, **Newton Masinde**, and Kalman Graffi. “Search Algorithms for Distributed Data Structures in P2P Networks”. In: *2018 International Symposium on Networks, Computers and Communications (ISNCC).* IEEE, June 2018. doi: 10.1109/ISNCC.2018.8530977

9. **Newton Masinde**, Gerald Schaefer, and Iakov Korovin. “Stable and reliable predictive accuracy of robust weighted slope one under profile injection attacks”. In: *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV).* IEEE, May 2016, pp. 1041–1046. doi: 10.1109/ICIEV.2016.7760157

10. **Newton W Masinde** and Sameen S Fatima. “Effect of varying filler-size in profile injection attacks on the Robust Weighted Slope One”. In: *Proceedings of the 2nd Pan African International Conference on Science, Computing and Telecommunications (PACT 2014).* IEEE, July 2014, pp. 92–97. doi: 10.1109/SCAT.2014.7055125

Students supervised

Master Theses

1. Moritz Kanzler. “Caching strategies for distributed data management”. Masters thesis. Düsseldorf, Germany: Institute of Computer Science, Heinrich Heine University, 2019.

2. Iakov Arkadjevic Dlikman. “Systematic Benchmarking of a fully Distributed Communication System”. Masters thesis. Düsseldorf, Germany: Institute of Computer Science, Heinrich Heine University, 2018

Master Projects

3. Moritz Gericke. *Secure Multiparty Computation in a P2P-based Online Social Network*. Institute of Computer Science, Heinrich Heine University, Düsseldorf, Germany, 2020.

4. Sebastian Bischoff. *Capacity Management Protocol for a P2P-based Online Social Network*. Institute of Computer Science, Heinrich Heine University, Düsseldorf, Germany, 2020

Bachelor Theses

5. Liat Khitman. "Benchmarking of a Distributed Framework for Online Social Networks". Bachelor thesis. Düsseldorf, Germany: Institute of Computer Science, Heinrich Heine University, 2020.

6. Philipp Mainz. "Introduction of a project management paradigm and analysis, redesign and development of a P2P-based Application Framework". Bachelor thesis. Düsseldorf, Germany: Institute of Computer Science, Heinrich Heine University, 2019

Interests and Extracurricular Activities

Reading, Basketball, Travelling.

Personal Publications

Articles

1. Kalman Graffi and Newton Masinde. “LibreSocial: A peer-to-peer framework for online social networks”. In: *Concurrency and Computation: Practice and Experience* (Dec. 2020). SPECIAL ISSUE PAPER, pp. 1–26. DOI: 10.1002/cpe.6150

2. Newton Masinde and Kalman Graffi. “Peer-to-Peer based Social Networks: A Comprehensive Survey”. In: *SN Computer Science* 1.5 (2020), pp. 1–51. DOI: 10.1007/s42979-020-00315-8

3. Newton Masinde, Liat Khitman, Iakov Dlikman, and Kalman Graffi. “Systematic Evaluation of LibreSocial–A Peer-to-Peer Framework for Online Social Networks”. In: *Future Internet* 12.9 (2020), p. 140. DOI: 10.3390/fi12090140

Reviewed conference papers

4. Newton Masinde, Sebastian Bischoff, and Kalman Graffi. “Capacity Management Protocol for a Structured P2P-based Online Social Network”. In: *Proceedings of The 7th IEEE Int Conference Social Network Analysis, Management and Security.* SNAMS 2020. IEEE, Dec. 2020, pp. 1–8. ISBN: 978-1-7281-7216-3

5. Newton Masinde, Moritz Kanzler, and Kalman Graffi. “Caching Structures for Distributed Data Management in P2P-based Social Networks”. In: *Proceedings of the 7th International Symposium on Networks, Computers and Communications (ISNCC 2020).* IEEE, Oct. 2020, pp. 1–8. DOI: 10.1109/ISNCC49221.2020.9297202

6. Raed Al-Aaridhi, Iakov Dlikman, Newton Masinde, and Kalman Graffi. “Search Algorithms for Distributed Data Structures in P2P Networks”. In: *Proceedings of the 5th International Symposium on Networks, Computers and Communications (ISNCC 2018).* IEEE, June 2018, pp. 1–8. DOI: 10.1109/ISNCC.2018.8530977

7. Newton Masinde, Gerald Schaefer, and Iakov Korovin. “Stable and reliable predictive accuracy of robust weighted slope one under profile injection attacks”. In: *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE, May 2016, pp. 1041–1046. DOI: 10.1109/ICIEV.2016.7760157

8. Newton W. Masinde and Sameen S. Fatima. “Effect of varying filler-size in profile injection attacks on the Robust Weighted Slope One”. In: *Proceedings of the 2nd Pan African International Conference on Science, Computing and Telecommunications (PACT 2014).* IEEE, July 2014, pp. 92–97. DOI: 10.1109/SCAT.2014.7055125

Theses

9. Newton Masinde. “Robust Weighed Slope One Algorithm”. Master’s thesis. Hyderabad, India: Department of Computer Science and Engineering, University College of Engineering (Autonomous), Osmania University, Oct. 2013

10. Newton Masinde. “Implementation of a Sequential Logic Counter”. Bachelor’s thesis. Nairobi, Kenya: Department of Electrical & Electronic Engineering, Jomo Kenyatta University of Agriculture and Technology, Apr. 2006

List of Figures

List of Tables

Eidesstattliche Erklärung
laut §5 der Promotionsordnung vom 06.12.2013

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

____
Ort, Datum

____
Newton Wafula Masinde

Please add here

the DVD holding sheet

This DVD contains:

- A *PDF* version of this thesis
- All LaTeXand grafic files that have been used, as well as the corresponding scripts
- The referenced websites and papers