# The Impact of Resource Sharing on Coexisting P2P Overlays and Stacked Overlay Modules

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf


vorgelegt von

**Tobias Amft**

geboren in

Tönisvorst


Düsseldorf, September 2017

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Jun.-Prof. Dr.-Ing. Kalman Graffi
Korreferent: Prof. Dr. Martin Mauve
Tag der mündlichen Prüfung: 23.10.2017

*Für Nessa...*

# Abstract

Different peer-to-peer (*P2P*) systems have been proposed in the last two decades, first investigated in research, then applied to applications and commercial systems. In contrast to traditional client-server approaches, P2P systems do not follow a strict centralized infrastructure paradigm. Instead, the functionality of one specific P2P system is distributed over all its participants fully or partially, depending on its purpose. Thus, the most outstanding characteristic of P2P systems is that they organize their own infrastructure and thus remain scalable even if they grow or shrink.

Peer-to-peer overlays which can be part or base of a complex P2P system introduce further logical layers on top of existing layers and extend their underlaying networks with own routing tables and forwarding strategies. The parallel execution and combination of multiple overlays with different and contrasting functionalities has been rarely considered in preceding research, although the combination and conjunction of specialized overlays seem to be a promising approach to reduce maintenance, implementation and execution costs of all overlays running on one single peer.

In order to motivate our research on P2P networks, especially on distributed online social networks and overlays for anonymous communication, we review political events which took place around 2010 and 2011 in the Middle East and Northern African region. We review modern techniques used by governments and oppressive regimes to monitor and censor Internet traffic and we discuss possible countermeasures against their attacks. Considering the uprisings during the Arab Spring and the related Internet shutdown events as they happened in Egypt and Libya in 2011, we find that distributed applications are highly demanded niche products among the increasing amount of services offered on the Internet.

In this thesis, we focus on the optimization of coexisting P2P overlays and applications operated in parallel on one peer. We identify basic requirements and characteristics of P2P overlays and applications and identify patterns which are frequently repeated, with the goal to avoid duplicate implementation and operating costs and to describe a methodology to create new overlays efficiently. As result, we present a novel approach to build coexisting overlays, named *overlay stack*, which is the idea to combine different overlay parts in a way that a desired behavior is obtained and second, that the implementation of duplicate modules and functionalities is avoided so that unnecessary expenses are kept at a minimum.

Following our goal, we focus on a contrasting set of overlays which differ in their requirements and functionalities so that core functionalities of P2P overlays and applications can be found. We divide the considered overlays into location-centric and social-centric approaches with restricted or free possibilities to communicate. With the implementations and evaluations of the proposed overlays algorithms in the four categories we show that a distributed hash table (*DHT*) or another indexing overlay constitutes a suitable basis for a diverse set of P2P applications. Moreover, we show that a DHT as basis is suitable to realize a contrasting range of P2P applications on top and identify one possible way to organize coexisting overlays on a single peer. The main benefit of a common DHT as basis for multiple overlays is that maintenance costs and vital update mechanisms are only spent once. By applying new applications on top of the common DHT and other existing overlays or applications, our overlay stack can be extended vertically with new functionalities.

Pursuing or goal to find core P2P functions, we present a new approach to enhance existing DHTs with social-relationship graphs according to Dunbar's *Social Brain Hypothesis*. Our approach enables trusted, friendship-based routing in structured P2P networks on the one hand and replication on trusted nodes with high data availability ratios on the other hand. We show that data in distributed online social networks can be kept available if few friends of a node are used as replication nodes only. Applications using the proposed social DHT benefit from the existing routing table and are able to use already known contacts for bootstrapping. In the field of location-aware services, we identify two classes of overlays and compare them in in a simulations-driven evaluation. Doing this, we show how mapping-based overlays can be realized on top of existing DHTs in the way that both, the location-based application and the underlaying DHT are decoupled so that they can be exchanged at any time. In the fourth category of investigated overlays, we show that during isolation, the ring structure in Chord overlays breaks up so that multiple separated communication islands are formed which are not able to re-structure without external help. To overcome this issue, we present our Ring Reunion merging algorithm which in case of isolation events is activated to re-organize corrupt overlay structures while keeping additional message overhead low.

Applying different P2P applications on a common DHT, we notice that one DHT defines a good core set of functionalities for different P2P applications, but is not sufficient to serve all demands of all possible applications, especially in cases in which more specialized routing mechanisms are required. To further reduce implementation and operating costs in DHT and other routing overlays, we search for a more general core set of overlay functionality and find out that P2P overlays mainly consist of three overlay modules which serve one specific purpose in the overlay. Routing tables are passive modules which give access to contact information to other modules. Routing algorithms are modules which implement forwarding rules and therefore shape the behavior of the overlay. Additional update mechanisms, like join operations or periodic keep-alive messages, are used to fill and update the routing table of an overlay. Introducing a common routing table, which serves other overlay modules, we are able to reduce further costs caused by duplicate modules and we allow to extend our overlay stack horizontally with new overlay functionalities.

In conclusion, we propose the overlay stack, a novel methodology to build coexisting overlays. We show that P2P applications in the overlay stack can be built on top of existing structures, and we show that new overlay functionalities can be added in form of specialized overlay modules on top of a common routing table. In addition, we show in our evaluation that the use of a shared routing table increases the robustness of all participating overlays while operating costs and implementation costs of the overlays are reduced. We further show that an overlay stack allows to switch on and off new routing functionalities during runtime, since the common routing table supports bootstrapping new overlays without the need of full join procedures.

# Zusammenfassung

Innerhalb der letzten zwanzig Jahre wurden verschiedene Peer-to-Peer-Systeme (*P2P*) vorgestellt, die zunächst akademisch motiviert waren und nun ihre Bedeutung für Anwendungen und kommerzielle Produkte zeigen. Im Gegensatz zu herkömmlichen Client-Server-Anwendungen folgen P2P-Systeme keinen strengen Vorgaben einer zentralisierten Infrastruktur. Stattdessen wird die Funktionalität eines Systems, je nach Zweck, sei es vollständig oder gar teilweise, auf seine Teilnehmer verteilt. Auf diese Weise können P2P-Systeme ihre eigene Infrastruktur bereitstellen, sodass sie skalierbar bleiben, selbst wenn Teilnehmer dem System beitreten oder es verlassen.

Peer-to-Peer Overlays können selbst Teil eines komplexen Systems sein und erweitern bestehende Netzwerke um eine logische Schicht, die eigene Routing-Tabellen und Strategien zur Weiterleitung von Nachrichten umfasst. Der parallele Betrieb und die Kombination mehrerer Overlays mit verschiedenen Eigenschaften und gegensätzlichen Anforderungen wurde selten untersucht, obwohl die symbiotische Kombination und Verknüpfung verschiedener Overlays einen erheblichen Beitrag zur Reduzierung von Laufzeit-, Wartungs- und Implementierungskosten leisten könnte.

Zur Motivation unserer Forschung an P2P-Netzwerken, besonders an verteilten sozialen Online-Netzwerken und Anwendungen zur anonymen Kommunikation, begutachten wir die politischen Vorkommnisse um 2010 und 2011 im Norden Afrikas und im Nahen Osten. Wir inspizieren moderne Techniken, die von Regierungen und unterdrückerischen Regimes zur Überwachung und Zensur von Internetverkehr verwendet werden, und diskutieren über mögliche Gegenmaßnahmen dieser Attacken. Betrachtet man die politischen Aufstände während des Arabischen Frühlings und die damit verbundenen Maßnahmen in Ägypten und Libyen, das Internet landesweit abzuschalten, so erscheinen verteilte P2P-Anwendungen eine Nische sozialer Kommunikation darzustellen, die gefüllt werden muss.

In dieser Arbeit betrachten wir daher die Optimierung koexistierender P2P-Overlays und Anwendungen, die parallel auf einem Host laufen. Wir identifizieren grundlegende Anforderungen und Merkmale von P2P-Overlays und Anwendungen und beschreiben Muster, die sich in vielen Anwendungen wiederholen. Unser Ziel dabei ist es, doppelte Implementierungs- und Betriebskosten zu vermeiden, sowie Methoden zur effizienten Erstellung neuer Overlays zu erarbeiten. Als Ergebnis präsentieren wir einen neuen Ansatz zur Erstellung koexistierender Overlays, den wir *Overlay Stack* nennen. Die Idee dahinter ist, verschiedene Teile und Module eines Overlays so zu kombinieren, dass ein gewünschtes Verhalten gebildet wird und zweitens, dass die Implementierung doppelter Module und Funktionen verhindert werden, um unnötige Kosten so gering wie möglich zu halten.

Um dieses Ziel zu erreichen, betrachten wir eine Menge an Overlays mit kontroversen Anforderungen und Funktionalitäten, sodass wir grundlegende Funktionen verschiedener P2P-Overlays und Applikationen kennen lernen. Wir unterteilen die betrachteten Anwendungen in positionsbasiert und sozial motiviert, als auch in frei oder limitiert, je nachdem, ob Kommunikation frei zugänglich oder eingeschränkt ist. Anhand der Implementierungen und Analysen der Anwendungen in den genannten vier Kategorien zeigen wir, dass eine verteilte Hashtabelle (*DHT*) oder andere verteilte Indexing-Verfahren eine grundlegende Basis für die Umsetzung verschiedenster koexistierender P2P-Anwendungen darstellt. Der große Vorteil einer gemein-

samen DHT als Grundlage verschiedener Anwendungen ist, dass Kosten für die Stabilisierung und Aufrechterhaltung eines Overlays nur einmal ausgegeben werden müssen. Der Overlay Stack kann vertikal erweitert werden, indem neue Applikationen auf die bestehende, gemeinsame DHT, bestehende Overlays oder existierende Anwendungen aufgesetzt werden.

Neben der Erforschung des Overlay Stack zeigen wir im Rahmen dieser Arbeit eine Möglichkeit auf, existierende DHTs mit sozialen Topologien nach Dunbar zu erweitern. Auf diese Weise ermöglichen wir einerseits vertrauenswürdiges und freundschaftbasiertes Routing in strukturierten P2P-Netzwerken und Replikationsmechanismen mit hohen Raten an Datenverfügbarkeit andererseits. Wir zeigen, dass Daten in verteilten, sozialen Online-Netzwerken verfügbar bleiben, wenn eine geringe Menge an befreundeten Kontakten als Replikaknoten fungieren. Anwendungen profitieren von der sozialen DHT während Bootstrapping-Phasen durch die Verwendung bereits bekannter Overlay-Kontakte. Im Bereich der positionsbasierten Overlays vergleichen wir zwei mögliche Klassen von Lösungen in unseren Evaluationen. Wir zeigen, wie Mapping-Verfahren genutzt werden können, um positionsbasierte Anwendungen auf einer austauschbaren DHT zu ermöglichen. In der vierten Kategorie verschiedener Anwendungen zeigen wir, dass die Ringstruktur in Chord während der Isolation eines Teils des Netzwerkes aufbricht, sodass kleine Kommunikationsinseln entstehen, die ohne externe Hilfe nicht wieder zusammenfinden. Dieses Problem lösen wir mit unserem *Ring Reunion Algorithmus*, der im Fall einer Isolation korrupte Overlay-Strukturen repariert und mit geringen Extrakosten zusammenführt.

Bei der Umsetzung verschiedener P2P-Anwendungen oberhalb einer gemeinsamen DHT stellen wir fest, dass eine DHT allein nicht alle möglichen Anforderungen beliebiger Anwendungen bedienen kann, vor allem dann nicht, wenn besondere Routing Verfahren von der Basis DHT gefordert werden. Damit wir Entwicklungs- und Betriebskosten weiter senken können, suchen wir ein eher generelles Konzept zur Erstellung neuer Overlay Funktionalität und finden heraus, dass P2P-Overlays hauptsächlich aus drei Modulen bestehen, die jeweils einen spezifischen Zweck im Overlay erfüllen. Routing-Tabellen sind passive Module, die anderen Modulen Zugriff auf Kontaktdaten gewähren. Routing-Algorithmen dagegen sind Module, die Regeln zur Weiterleitung von Nachrichten implementieren und dadurch das grundlegende Verhalten eines Overlays beschreiben. Zusätzliche Update-Mechanismen, die den Beitritt in ein Overlay ermöglichen oder periodisch Keep-Alive-Nachrichten versenden, füllen die Routing-Tabelle und aktualisieren sie. Durch die Einführung einer gemeinsamen Routing-Tabelle, die verschiedenen Modulen im Overlay Stack zur Verfügung steht, können wir weitere Kosten vermeiden, die vor allem durch redundante Module entstehen. Weiterhin erlauben wir durch die modulare Entwicklung eine horizontale Erweiterung des Overlay Stacks durch neue Overlay-Funktionen.

Zusammenfassend beschreiben wir den Overlay Stack, eine neue Methode zur effizienten Realisierung koexistierender Overlays. Wir zeigen, dass P2P-Anwendungen im Overlay Stack auf der Basis existierender Module gebaut werden können. Außerdem zeigen wir, dass neue Overlay-Funktionen als neue, spezialisierte Module auf einer gemeinsamen Routing-Tabelle gebaut werden können. Unsere Evaluationen zum Overlay Stack beweisen, dass der Gebrauch einer gemeinsamen Routing-Tabelle zusätzliche Robustheit aller Overlays im Overlay Stack mit sich bringt, während die Betriebskosten der implementierten Overlays reduziert werden. Besonders interessant ist, dass unser Overlay Stack die dynamische Zu- oder Abschaltung neuer Funktionalität zur Laufzeit erlaubt. Dies ist unter anderem möglich, da die gemeinsame Routing-Tabelle eine Menge an Overlay Kontakten für Bootstrapping-Phasen bereit stellt, ohne dass ein Overlay vollständig dem Netzwerk beitreten muss.

# Acknowledgements

An dieser Stelle möchte ich einigen Menschen danken, die mir während der Anfertigung der Dissertation wahnsinnig geholfen haben.

Kalman Graffi und Martin Mauve danke ich für ausgezeichnete fachliche Betreuung während der Promotion. Besonders die Motivation und Begeisterung die Kalman Graffi in unzähligen Diskussionen und Gesprächen aufgebracht hat, waren mir eine gute Lehre.

Ich möchte außerdem meinen Kollegen der Arbeitsgruppen "Rechnernetze" und "Technik sozialer Netzwerke" für eine lustige und entspannte Arbeitsatmosphäre danken. @Thomas, Sabine, Angela: ihr seid unsere Helden im Hintergrund. @WeGi: danke für die Bilder !!! @Krauti: "You motivated me awesome!" @Meter: Dir danke ich für deine langen, penetranten Vorträge über Clean-Code, TDD und funktionale Programmierung. Du hast mir eine Menge zu (über-) denken gegeben und neue Denkweisen aufgezeigt. @Schweinemensakapelle: mit euch war es "leider geil".

Den Studentinnen und Studenten, die ich während der Promotion bei ihren Abschlussarbeiten betreut habe, danke ich für ihr Vertrauen und für viele spannende und auch amüsante Diskussionen.

Zu guter Letzt danke ich meinen Freunden und meiner Familie, die mich auf dem Weg der Promotion begleitet, unterstützt, und ertragen haben.

Ohne euch wäre ich nur halb so vollständig wie ich bin...

Herzlichen Dank

# Contents

# Chapter 1

# Introduction

Moore's law says that the number of transistors in an integrated circuit roughly doubles every one or two years. Although detractors doubt that this law would hold for long time, one can draw an important observation out of it: complexity and functionality in modern technology are growing rapidly. The evolution of technical devices proceeds without halt, leading to plenty of future trends which could have been promoted in any science fiction novel written in the past. During the last centuries, lots of efforts were put into communication techniques and equipments to share information over long distances. When in the 1850s, the first transatlantic telegraph cables were installed to connect America with Europe, new possibilities of communication were born. Message could be sent via copper wire across the Atlantic Ocean in an instant, whereas ships needed ten days to cross the ocean. Almost hundred years later, in the late 1960s, the ARPANet (*Advanced Research Projects Agency Network*) started as small scientific communication network between four universities in America. The ARPANet evolved steadily until it was released in 1990 for commercial purposes as an early version of the Internet. What started as small experiment, grew larger to a global network, known as the Internet.

By now, billions of devices are connected to the Internet. Current developing trends like Industry 4.0 and Internet of Things (*IoT*) predict even more devices to be connected in the following years. Whereas decades ago, most people received their knowledge from books or even encyclopedias stored on CDs, nowadays one simply asks her/his smartphone or any other smart device to search the web for an answer. Information then travels around the earth, almost at the speed of light, available at any time of the day. The Internet as it exists today influences many lifes in different ways since it allows to share information, communicate, or discuss with people all over the world without deep technical knowledge. Thanks to upcoming trends in *virtual reality* (*VR*), it is even possible to take part at events and activities located far away. As a result, online social networks (*OSN*) like Facebook, Twitter, Weibo, and many others became prominent communication platforms in the last years which are expected to grow even <further in near future. Increasing numbers of users participating in online social networks and other Internet services, as well as the expanding mass of devices being connected to the Internet, force operators of those services to react and to extend their hardware and infrastructure continuously. Additionally, combinations of web services and smart marketing models like integrated advertisements keep the costs for users low.

On the contrary, the high participation of users in Internet services allows organizations, agencies, governments, or regimes to spy upon/overhear traffic and conversations. Egypt's Internet shut down as reaction to uprisings during the Arab Spring (2011), the NSA affair around

Edward Snowden (2013), as well as recent attempts of the Turkish government to block services like Facebook, Twitter, WhatsApp (2015, 2016, 2017) are all evidence that operators and authorities are interested in controlling Internet traffic. Without further evaluation it is conceivable that social interactions as well as controlling facilities are more and more shifted into the Internet, which is far away from being an exhausted market, since it is continuously growing.

## 1.1 Motivation and Problem Statement

Most services running on the Internet use a client-server architecture. Typically, a remote machine called server offers one or more services and listens to incoming connections from clients which are consumers of those services. In that way, servers respond to incoming requests from clients. The amount of clients a server can handle simultaneously is limited by its connection to the Internet, provided by an Internet Service Provider (*ISP*). Thus, client-server approaches and corresponding applications usually do not scale with increasing network size. Recent events show repeatedly that the infrastructure provided by the client-server approach is an easy target to control and censor its users. Since the Internet is divided into many autonomous systems which are controlled by different entities like authorities or companies, it is very easy for governments, agencies, and attackers to intercept and block traffic in the Internet. The Turkish government blocking Twitter access, China with its *Great Firewall*, or the global surveillance program of the NSA revealed by Edward Snowden are only few examples of active manipulation of Internet traffic.

Distributed peer-to-peer (*P2P*) protocols became popular around the year 2000 and are in focus of researchers for almost two decades now. They define an interesting alternative to the client-server approach, because their distributed infrastructures make it difficult for attackers to control, monitor, or block the system, because no central control instance exists. Well known representatives of these protocols are file-sharing applications like Napster [121], Gnutella[1], Bittorrent [35], or Kademlia [94]. In contrast to client-server approaches, peer-to-peer networks do not follow a strict and fixed infrastructure that depends on a central server. Instead, participants, also named peers, actively take part in the organization of the system. Depending on the protocol, they are equal in their roles. In some ways, peers can implement both the client as well as the server side of a protocol. Important is that each peer only serves and knows a subset of other peers in the system. As a result, peer-to-peer networks and associated applications remain scalable if networks grow. A significant issue of P2P systems is to maintain the decentralized organization of the network under dynamic change of peers participating in the network. Since peer-to-peer applications define new functional networks above an existing network, for example the Internet, they can be classified as overlays. Speaking about peer-to-peer overlays, an overlay is the introduction of a new logical layer on top of an existing network. Moreover, the overlay extends the underlying network, the underlay, by routing structures and connections between peers, so that new routing functionality emerges.

Many divergent P2P overlays have been proposed in the last two decades, with file-sharing applications being the most prominent example. Other prominent application scenarios are distributed online social networks [72], distributed storage and data structures [5], decentralized

[1] http://rfc-gnutella.sourceforge.net/

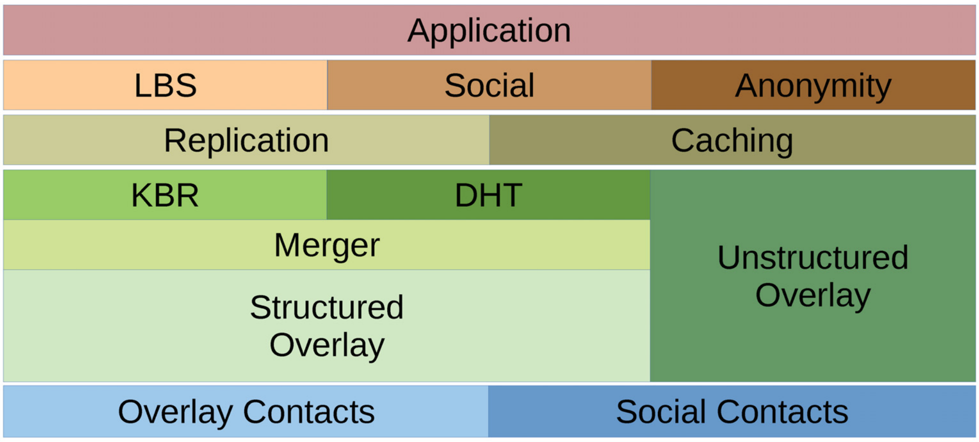


Figure 1.1: A schematic representation of the overlay stack. Each module exists only once in the stack and serves one specific purpose. Duplicate functionalities and modules are avoided.

lookup services [50], distributed monitoring approaches [69] [51], and many others. We believe that in the next years, more and more services and applications are placed into the Internet. We also expect distributed P2P applications to become more meaningful, due to current efforts of browser vendors to prepare basic P2P functionality in their browsers. Out of these efforts, the WebRTC open project emerged which is a collection of useful P2P protocols, like STUN, TURN, ICE, etc., allowing different P2P protocols to be embedded in a browser, like Disterhöft et al. show in their work [52].

Striking point is that current overlay implementations only serve one specific purpose. The combination and parallel execution of those systems is certainly possible, but has rarely been considered or deeper investigated. Running different peer-to-peer overlays in parallel could be very inefficient, especially if multiple overlay realizations share the same functional requirements, but are implemented separately. In contrast to the naive approach of running different overlays in parallel, the introduction of different layers and modules which can be stacked together might lead to better performance. Important functionalities which are implemented multiple times, should be comprised in a single module only, so that any costs which come along with different functionalities are reduced to a necessary minimum. The resulting overlay stack, which is presented in a visualized form in Figure 1.1, consists of different basic modules which provide basic overlay functionalities. The bottom of the stack represents the heart of the overlay stack. It consists of basic routing mechanisms which provide search and lookup functionality. From bottom to top, the modules become more and more application-like, so that the top of the stack comprises application modules which serve one specific purpose. The stacked approach allows to exchange individual modules or to add new overlay functionalities easily in further modules. Up to now, the creation of overlay stacks as well as the interdependencies and requirements has not been studied and is therefore not fully understood. Deeper insights into the concept of the overlay stack are given in Section 1.3.2 and in Chapter 8.

In order to be able to investigate the quality, interdependencies, and requirements of different modules in one overlay stack, several types of dedicated overlays need to be explored, that differ in their purpose. To reduce the amount of dedicated peer-to-peer overlays considered for deeper investigation, we limit our examination to four types of overlays and applications. The four overlays selected are on the one hand interesting topics with regard to current trends in Internet services, on the other hand they are selected to be contradictory as they have competing restrictions and requirements. The overlays are divided into location-centric (L) as well as social-centric (S) overlays. Additionally, we classify protocols in restricted (R) or free (F), according to the way communication is allowed or restricted in the specific overlays. With this classification, we cover divergent and opposing use cases and we investigate overlay stacks from different points of view.

**Social-centric, free communication:** Online social networks (*OSN*) are services or applications provided in the Internet which enable its users to act socially in an online virtual world. Social interests and interactions, like sharing data or knowledge, presenting an own website or online profile, or using different communication channels to stay in touch with companies, family, and friends, can be moved to online social networks. The user mostly benefits from easy usage of those services. Main problem in current OSNs is that users lose control over their data, as they give it to a remote service provider that often has its own terms of use. In contrast to client-server based OSNs, distributed online social networks (*DOSN*) earned a lot of attention in the last years. DOSNs implement online social networks in a distributed manner with an underlying P2P overlay as basis. In this way, DOSNs can be operated without the need of central servers that organize the service infrastructure. Instead, users of DOSNs maintain their own infrastructure and organize their data in a distributed way. Recent events like the Arab Spring as well as the following blocking of online social networks foster users' wishes to gain more control over their data and to manage the censorship resistantly. Two approaches for DOSNs emerged as reaction. First, using the private server approach, users set up their own hardware and keep their own data and possibly data from friends on own servers. In this way, the control remains at the users and data is located near a user. Second, pure P2P overlays have been introduced, which take advantage of social structures and organize their routing tables in the way that social networking is fostered. Administration reaches in both approaches from one to one communication over group communication up to to n-to-n communication, also called publish-subscribe. Challenging in P2P overlays and especially in DOSNs is the frequent joining or leaving of participants (*churn*). Whereas centralized servers are available at any time, peers in distributed networks could leave at any moment, causing the overlay to become unstable. The principal question is where data should be stored in pure DOSNs. Different approaches exist to reach a maximum of data availability, for example by storing data at friendly nodes which are assumed to be online for long times. In Chapter 4 and in our paper [72], we present a DOSN which offers a data availability service based on trusted nodes.

**Social-centric, restricted communication:** For many centuries now, people wish to communicate in private so that others are not able to overhear classified messages. Nowadays, people still feel the urge to communicate anonymously and in private. Surely, the Internet makes it possible to post personal messages in an OSN, Blog, or Forum. People's real identities are hidden up to a certain degree. At least other users are not able to identify a user directly if a nickname is used, but providers of OSNs are able to track communication and to associate users with their accounts. In some parts of the world, this knowledge about who is talking with whom is enough to punish people and accuse them for espionage. Erdogan's persecution of alleged Gülen-followers after the failed military putsch is one recent example of people being monitored or arrested because they are assumed to be in contact with a suspected person. One should also be aware that privacy and anonymity are two distinct topics. Communication is said to be *private*, if nobody else is able to identify the matter of a talk i.e. no one knows what participants are talking about. This goal can be achieved easily with encryption. *Anonymity* on the other hand is the ability to hide sender, receiver, or both from each other or better from a third spectator. P2P overlays in their decentralized way provide some features for anonymous communication per se, since no controlling instance like a centralized server exists. There is normally no server known to the public which is called/requested to pull data from. Lookups or searches require under normal conditions multiple hops to deliver a message between two peers. As a result, messages traverse from its initiator to a target peer on a path which is not known to participants. Each participant has only local view onto the overlay. The message path is hidden so that normally, users could only be tracked by an observer who controls the underlay and is therefore able to track messages. Since the Internet consists of many autonomous systems, and autonomous systems are normally controlled regionally, it is uncertain that a global viewer is able to capture all traffic at once. Another challenge in P2P is to hide information from other participants. Since most overlays use the Internet as underlay, IP addresses are exposed to other peers. Existing approaches for P2P overlays which foster private and anonymous communication can be classified into three categories: friend-to-friend (*F2F*), group-based, or anonymous P2P overlay. Whereas in friend-to-friend approaches only friends are connected, in group-based solutions, a group key is created with which communication can be encrypted and hidden from peers outside the group. In the third category of anonymous P2P overlays, peers are connected to few unknown, other participants. Problems of all P2P approaches are on the one hand the flexible user participation which leads to instabilities in the overlays, and on the other hand limited knowledge about other participants which reduces the possibilities to route messages in the overlay. Still it is unclear in how far the different approaches support anonymous communication. Clearly, in all three approaches, communication between peers is restricted to a certain degree. In Chapter 5 and in our paper [11], we show how distributed hash tables (*DHT*s) can be extended with social-relationship graphs, so that routing via trusted nodes is enabled.

**Location-centric, free communication:** Location-aware in the context of P2P overlays means that peers are associated with a physical location of which they are aware. Goal of a P2P overlay in this class is to order and connect peers according to their location so that location-dependent objects can be managed and found. The motivation for distributed location-aware applications emerged from the wish to reduce costs for an underlying infrastructure. The use cases for those applications are various: tracking service, restaurant search, area search, geographical storage, and many others. The idea behind location-aware overlays is either to associate data with certain locations and store them inside the network, or to find peers in the network which have a certain position. Whereas location-aware client-server applications/services are very easy to realize, the distributed P2P alternative is more challenging. The challenge mainly lies in finding the right approach to fulfill the needs of the overlay. Two approaches are mostly used in this class of overlays: either nodes are mapped to geographic locations or locations are integrated into a peer's status. In the first case, locations are mapped to peers in an already existing overlay. As an example, space-filling-curves or other linearization approaches could be used to map peers in Chord [133] to geographic positions. In this first group, peers do not necessarily have to be located in the region they are responsible for. Instead they manage nodes located in this position. Advantage is that DHTs and many approaches can be reused as basic overlay. Furthermore, positions of super peers, those responsible for a specific area, can be calculated directly so that a lookup for data or peers is possible. Disadvantage is, that for range-queries and area searches multiple nodes have to be visited. Also, peers might be close in Id Space but far away located, or vice versa. In the second case, nodes maintain routing tables which are filled with nodes located within a certain neighborhood. In this case, nodes associated with a location and are directly connected to neighboring nodes. Challenge in this case is to maintain links to close and distant neighbors. Distant neighbors are important to reach far places fast. Contrary to the first approach, an area search is a search in which all nodes close to a desired location are visited until all desired nodes/or data objects are found. The problem is that those approaches suffer from fix routing table sizes. Routing tables of solutions in the second case usually do not grow with the network size. Thus, it might happen that an area search is not deterministic, depending on which nodes are searched. Years ago, when most P2P overlays were introduced as research topic, computers were rather fixed to one location. As a result, location-aware overlays developed in this time mostly considered stationary peers. Now, more than one decade later, computers are mobile and not longer fixed to one place. Smartphones, smartwatches, etc. are small enough to be carried around. Most of them have GPS functionality e.g. for sports or navigation purpose. Through the mobility of peers, new challenges for location-aware P2P overlays arose. Most challenging is to keep connections to other peers up to date. In Chapter 6 and in our paper [8], we introduce a location-aware P2P overlay which is aware of peer movement and compare our approach to another existing solution.

**Location-centric, restricted communication:** In this classification, partitioning-aware overlays are investigated. More specific, partitioning-aware overlays and services are able to react on sudden connection losses that occur in the underlay. Characteristically, during a partitioning or isolation event, a whole fraction of users loses simultaneously connection to other participants in the network. As a result, different communication islands are formed which are not interconnected, but allow communication between participants inside the islands. Leave and join of participants happen frequently in P2P networks, a protocol should be aware of this fact and should be able to stabilize its functionalities. as reaction to churn. Rarely considered in literature is the sudden connection loss of huge groups of users. It might happen though, that an overlay is ripped into two or more parts, so that the smallest partition holds up to 50 percent of all users. Under normal conditions, in real world, participants of P2P networks are spread all over the world. They are connected through their Internet service providers (*ISP*s) to the Internet. The Internet itself consists of several autonomous systems which are connected with the help of interior gateway and exterior gateway protocols. Normally, an ISP is associated to one country and is therefore bound to the telecommunications law of that country. In some cases, governments take sterner actions to control the Internet traffic in their country. Best example are the northern African uprisings around 2010 and 2011. Social media played a great role during this time, since people used that media to speak about the uprisings, to share and express their feelings to the outer world. In Egypt, where most people shared the political situation via Facebook, Twitter and YouTube [132], the regime at this time forced/instructed the local ISPs to shut down their services so that communication to other (western) countries is restricted. In fact, they deleted routes in the border gateway protocol (*BGP*) connecting Egypt ISPs to other countries. As a result, Egypt was isolated from the rest of the world, but many hosts were able to communicate inside Egypt. The isolation of whole geographic areas leads to a partitioning in the Internet, which also affects overlays running above. As a result, one would expect that two partitions are formed during an isolation event, in the case that only one country is isolated. We found out that depending on the overlay, multiple communication islands are formed during isolation. Less spectacular is the following case: an overlay for social purpose is split into different groups or starts with different groups of

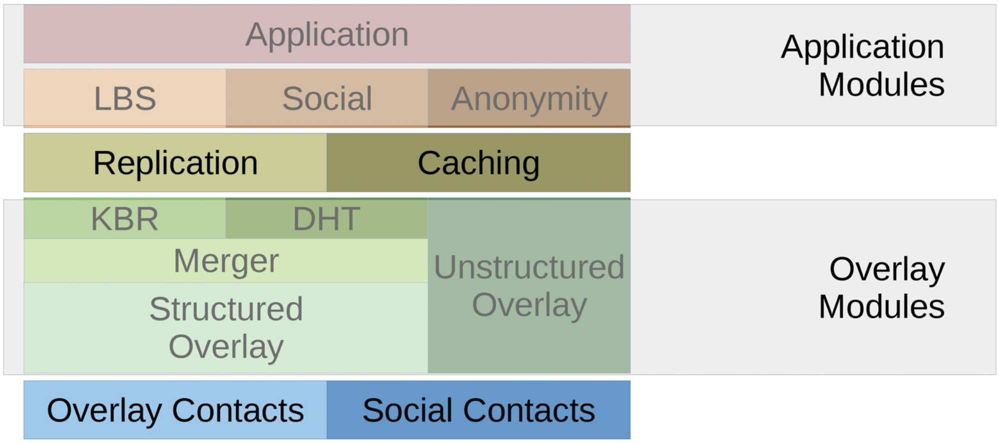


Figure 1.2: A schematic representation of the overlay stack. The application specific parts, as well as the overlay specific parts are highlighted.

participants. For each group one overlay exists. After some time the groups, for example fan-club, interested persons, etc., want to combine themselves. The challenge is now to create or extend overlays, in the way that they are able to merge different separated groups at any time. In the example of isolated countries, one would wish the following behavior: split groups and islands outside the country which are not separated should merge directly, as they have fully connectivity among themselves. Also, if communication within the isolated country is possible, for example because only BGP entries are deleted groups inside should find themselves so that after a while two distinct groups are formed. As soon as the state of isolation is finished, the groups should automatically merge again. The *Ring Reunion Algorithm*, our solution to this problem is presented in Chapter 7 and in our paper [7].

Part of this thesis is to develop solutions for each of the four overlay types. The resulting overlays are very important to understand, because they are possible parts of an overlay stack as it is described in Figure 1.1. We aim to realize the four approaches as modules on top of a common basis DHT, so that the costs for duplicate lookup services can be avoided. With this concept, new applications can be added as specialized modules on top of the common DHT. In Section 1.3.2 we further discuss the role of the four applications for our overlay stack.

Although speaking of overlays and applications in similar ways, we have to differentiate between both terms. Overlays are networks placed upon existing networks in order to provide new routing functionality. Doing this, further routing tables are maintained so that routes between two or more peers are found easily and sometimes faster than using the underlay. Moreover, an overlay is a distributed lookup service which finds routes to desired targets in the network through distributed organization of its participants.

An application typically implements a specific purpose upon the routing overlay. More specific, the application utilizes the routing mechanism of the overlay to find peers which are participating in the application. Actions performed at a peer which are not part of the overlay, namely routing table, routing algorithm, or updates mechanism, are considered to be part of the application. Basic examples are storage systems, file-sharing applications, and social network applications. Although both terms are different in their meaning, they might be used as one, since most application implementations are integrated in the corresponding dedicated overlay solution. The placement of overlay modules and application modules inside the overlay stack are shown in Figure 1.2

## 1.2 Research Questions

In the following, we discuss the research questions which are derived from the problem statements described above. Our main goal is to gain a better understanding on creating stacked overlay networks with desired functional and non-functional requirements and to provide a systematic approach and guideline for the creation of new overlays or single parts of an overlay. The collected insights deepen our understanding about coexisting overlay modules being assembled to an overlay stack, as it is sketched in Figure 1.1 and described in Section 1.3.2. It should be possible to extend the resulting *overlay stack* at any time with further modules for any overlay functionality.

First question of our research is how overlays with regard to the four diverse classifications can be constructed. The implementation of each single solution needs to fulfill at least two requirements. First, this thesis should clarify, which solutions for the presented use cases exist, how they differ, and what the characteristics of a single solution are. Second, we are interested in how far single solutions can be realized with respect to a possible overlay stack. In specific, even if solutions exist for any of the presented cases, we have to find out if and how these solutions can be integrated into an existing overlay stack or how they could be converged to a new basis overlay. Whether we could reuse and extend existing solutions or whether we create new approaches, the following questions arise:

- How can distributed online social networks be implemented on top of a DHT, especially, if availability of data has to be assured during churn? Although many approaches for different OSNs and DOSNs exist, they share almost equal functionality. Mainly they provide methods and infrastructures to enable the storage and retrieval of documents and data items, like profiles, walls, or blogs, in the network. Robin Dunbar, an anthropologist and psychologist introduced the Dunbar number which describes a limiting amount of social connections a person can normally maintain. We are interested if Dunbar's idea of different affinity groups which describe the communication intensity between human beings can be adopted to create a trust system for DOSNs and a respective replication mechanism, which are both based on ego-networks.

- We are interested in how far existing DHTs have to be extended, so that social-relationship information can be integrated into DHTs and structured routing algorithms in specific. Is it possible to integrate Dunbar's concepts into existing DHTs? In special, we want to know in how far anonymous, DHT-based overlays are possible to be realized if routing is restricted to trusted, friendly nodes only. How many contacts are needed to realize routing over friendly nodes only? Can the property of anonymity in DOSNs be transferred to other overlays? Is there a key set of functions, that if implemented, provide anonymous communication options?

- Which approaches of location-based P2P overlays exist and how do they differ from each other? Which general techniques do we have at hand to support mobility in location-based overlays? How can these core mechanisms be described and can they be added to other overlays to support mobility also in them? Is a common DHT sufficient enough as basis for a lightweight implementation of a location-based system which can be reused by other applications?

- Are there any approaches which consider restrictions in the underlay, especially partitioning events on large scale? How many routing table entries are outdated after a partitioning event and how many entries have to be restored to guarantee routing mechanisms to work reliably again? How does a merging algorithm look like, whose purpose it is to merge separated communication island during and after isolation? Which additional costs arise through the use of a merging algorithm, and how are those costs related to the efficiency of the merging algorithm?

In general, for each overlay in the four classifications we are interested in related state of the art concepts to find out which main requirements the application specific overlays constitute. Second, we are interested to apply the four applications on a single DHT which is used in a shared manner by the four solutions. The benefit of a common DHT is that duplicated costs for maintenance can be avoided and known contacts can be offered to any application.

Next, we are interested in a deeper investigation of the four application specific overlays and its characteristics. We want to find out what are basic requirements that a hosting overlay must provide in the context of the four applications. To answer this question, we try to identify basic requirements for different overlays and therefore focus on the functionalities and components each type of overlay provides. Hopefully, the identification of common behavior in solutions of one class leads to knowledge about the basic overlay structure serving a specific purpose. As a result we should be able to identify a core set of methods needed for a basis overlay. Furthermore we would like to know:

- What are basic requirements and characteristics of an overlay? Can we identify common patterns in existing overlays that are frequently repeated?

- Which functionality is application specific and which functionality is essential in the basic overlay?

- Is it possible to implement the different applications on top of an exiting DHT? Is it possible to implement an underlying DHT as common basis? Are there cases in which an underlying DHT as common basis is not sufficient? Which additional functionality would be needed to extend the DHT as basis?

- Are the requirements towards the underlay the same for all types of overlays, or does any overlay need special underlay extensions?

Based upon the knowledge about single solutions for dedicated overlays, we answer the questions, what are overlays made of and can we benefit of combining the essential elements of the individual overlays to a stacked overlay, which executes the individual solutions in parallel instead of individually. Most important for our work, knowing the functionalities and purposes of different overlays, we identify necessary components of an overlay. In special, we want to find out which parts of different overlays are equal. Question is if we can observe similar patterns and functionalities that are repeating and if we find such patterns, can we derive a basis for overlays? Additionally, we test if basic parts of overlays can be reused partially for other overlays.

Having identified a suitable approach to stack different overlays together, we are interested in possible drawbacks and side-effects of our solution. Having various overlays running in parallel in a stacked overlay framework might lead to horizontal interdependencies between the modules in the same layer. A merging algorithm for example, could influence other applications in a positive way by updating the routing table of the shared DHT lookup service. On the other hand, vertical interdependencies between different layers could arise. A possible scenario leading to vertical interdependencies could be a merger which is operating in one of two parallel executed DHTs. If both DHTs share one routing table, as described in Chapter 8, the DHT without merger and all applications built on top of it, benefit from the merger as well. Further questions emerged from the previous:

- In how far are single components of the overlay stack linked to other components? Are there dependencies between components, do they influence each other?
- Which interdependencies are necessary to form an overlay? Or are there any characteristic dependencies between different modules which cause a certain behavior?
- How is it possible to run different modules in an overlay stack? Can single modules in the stack be switched on or off? Do certain modules effect the behavior of a protocol if switched off and what is the effect?
- In how far is it possible not only to run overlays above others but rather aside others? Speaking of overlays, only the stacking is considered. What is the behavior of overlays running next to each other on the same level?
- We try to identify redundant functionality in different layers and investigate its effects and want to know: are there measurable effects of redundant functionality e.g. in maintenance operations?

As last step, we are interested in the costs and limitations of stacked overlays. It has to be cleared how costs for overlays on top of a common basis are measured and which overhead is produced. In the end we want to know at which costs different approaches of an overlay stack can be realized.

- We identify costs and limitations of stacked overlays.
- We identify trade-off of stacking overhead and flexibility. We want to know: is it better to provide most functions as basis or provide only core?.
- We compare overall quality of stacked functional layers to efficient single layer solutions.
- Most important, we want to find out which costs can be reduced through the use of an overlay stack and we want to know if the proposed cost reduction affects the quality of the overlays and applications we are investigating.

Finally, we should be able to comment on the question whether an overlay stack is a promising approach to reduce overhead emerged trough parallel executed overlays or whether dedicated solutions are more robust without underlying basis. We clarify, which cost optimizations are possible in an overlay stack and we investigate if dedicated solutions for different applications are generally more performant than solutions added on top of a common basis.

## 1.3 Contributions

The amount and complexity of Internet services is fastly growing in the last years. Contrary to traditional client-server based services, P2P overlays come in focus as decentralized alternative. Goal of this thesis is to categorize/investigate/summarize requirements, functionalities, and realizations of diverse and contrasting overlays. The conclusions about single overlay applications are transformed into the description of an overlay stack. The stacking of overlays is our approach to formulate rules for the combination of different P2P overlays so that existing implementations can be reused and redundant processes are reduced. With the description of an overlay stack, different features and applications are implemented in small, lightweight modules, each fulfilling one special purpose. The idea is to combine different modules to obtain a specific behavior of an overlay. Modules can be exchanged at any time to vary the system, new applications can be added simply utilizing existing modules. In this section, we describe contributions that are made to reach the goal of a description for overlay stacks step by step.

Recent events have shown that oppressive regimes exist that adopt drastic actions to hinder people to communicate freely in their country. The uprisings during the Arab Spring are a prominent example that online social networks are favored communication channels to organize groups and to share opinions with other people. The Chinese Republic which blocks and filters Internet traffic routinely is another example of Internet censorship. Since a big part of this thesis is to investigate distributed online social networks as well as anonymous networks, and many people in censorship affected countries feel the urge to communicate freely, we contribute in this thesis with a broader review of technical possibilities to censor Internet traffic. Additionally, we motivate and introduce distributed approaches and solutions against Internet censorship. We present an overview of current state of the art in distributed communication overlays in Chapter 2. Furthermore we value the transformation of online social networks and communication tools, as well as possible changes in computer science after the uprisings in the Middle East and Northern African region (*MENA*) in our paper [10].

To answer the question how overlay stacks in the context of the contrasting distributed applications could be created, we implemented different P2P protocols in a simulator. The choice of a simulator depends on multiple requirements of the simulated object. Since diverse applications emerge for different purposes, a simulation environment should be flexible enough to support agile developments. Small part of this thesis is to compare the similarities and contradictions of P2P networks, opportunistic networks (OppNets), and mobile ad-hoc networks (MANETs) in order to find a suitable simulator, that can be used to cover different but relative research topics about routing strategies in computer networks. Our contribution in Chapter 3, which is based on our paper [32], is to present a collection of simulation tools for current state of the art in P2P, OppNet, and MANET research. Further, we identify PeerfactSim.KOM[2] as best

[2]http://peerfact.com/

suited simulation environment to implement our protocols and to conduct our evaluations.

### 1.3.1 DHT-based Peer-to-Peer Applications

Various P2P applications have been proposed in the last two decades and many concepts have found their way into industry or society. Normally, each realized P2P application fulfills one specific purpose. Filesharing P2P networks became popular with the purpose to share data, like music, literature, pictures, etc., among millions of users. Most important functionality was to search for content. With BitTorrent [35] and similar content distribution concepts, the delivery of large data files has been improved and costs deployed to the users. Structured P2P networks have been proposed to provide efficient routing in P2P systems. Content is retrieved by identifying a responsible node to a given overlay identifier in a DHT. In this thesis, we extend the variety of existing overlay applications with further solutions and evaluations of these solutions.

Our goal is to derive construction rules and basic requirements for different overlays in order to identify common mechanisms and schemes for a variety of challenging use cases, so that we can provide an approach to support the creation of overlays with desired functionalities and quality properties through a supportive framework, such as an overlay stack Thus, we focus on various scenarios that define competing restrictions and requirements. Doing this we are able to identify similarities for a diverse set of overlay approaches. Regarding the four use cases presented in Section 1.1, our contribution comprises the design, implementation, and evaluation of solutions for each use case, realized as application or protocol on top of an existing DHT. The contribution of each use case is described in the following:

- **Social-centric, free communication:** In the class of social centric overlays, our contribution is the presentation of a novel P2P distributed online social network. The novelty here lies in the combination of approaches from social sciences with P2P overlays. The psychologist Robin Dunbar proposed the Dunbar Number as a measure to describe the amount of interaction between different individuals. According to Dunbar's *Social Brain Hypothesis* [55], the human brain is able to maintain only approximately 150 social contacts at once. Dunbar further divides these contacts into four affinity groups that describe the current social distance to other individuals. More specific, the composition of an individual ego and its maintained contacts is called ego-network and describes the interaction frequency between individuals. In Chapter 4 we include this concept of ego-networks in our work to build a DOSN which is able to provide typical social network functionalities like one-to-one communication, file storage, and others. Our goal in this work is to keep data available in the system even if nodes continuously join and leave and to optimize the overlay topology with regard to the expected user-to-user interactions. Our approach is to store data on trusted peers taken from a nodes ego-network. An underlying DHT is used to index peers in the system on which responding data is stored. Our evaluation reveals that ego-network-based DOSNs provide effective methods for data-availability services so that, like presented in our paper [72], not more than 2-3 replicas have to exist to guarantee data availability rates above 95 percent.

- **Social-centric, restricted communication:** We extend the concept of Dunbar-based networks further in Chapter 5 to create anonymous communication, which is the ability to hide sender, receiver, or both from each other and from any third person. In Chapter 2, we discuss the need of anonymous and private communication tools and review the technical realization of distributed P2P applications for this purpose. Further contribution is the proposal of a method to route messages between multiple ego-networks. The special challenge in overlays, that adopt ego-network connections in their routing tables is that connections to other peers are unidirectional. In most cases, a peer $p$ maintains a connection to peer $q$, but $q$ does not maintain a connection back to $p$. We further discuss the utilization of ego-networks for anonymous communication and show why a naive integration of Dunbar-based solutions into existing DHTs is not possible. With our paper [11], we enhance a regular DHT with social-relationship entries, so that DHT lookups can be performed with respect to forward messages among trusted nodes only.

- **Location-centric, free communication:** The existing solutions in the class of location-aware overlays are extended by our presentation of an lightweight solution for location-aware overlays on top of existing DHTs. In Chapter 6 we show that it is possible to build overlay functionality on top of a DHT which allows to associate peers in the network with their physical location. Participants of the system are able to search for other participants located in a specific geographic area. We compare our approach to another class of location-aware overlays in which peers cannot be found deterministically. We discuss advantages and disadvantages of different approaches and evaluate our approach under the assumption that peers are constantly moving. Most previous solutions in this class of overlays assume fixed locations of the participating peers. LobSter, our solution presented in [8] on the contrary is aware of moving peers. Area searches within a radius of approximately 4 percent of the simulated world produce the best result in terms of the ratio found nodes to total nodes in the area, and within this radius, the costs of LobSter in terms of visited nodes during search are comparable to other specialized solutions.

- **Location-centric, restricted communication:** Although churn is discussed in many P2P overlays, large-scale isolation events during which a great part of the underlay lose connection to the remaining part. Those partitioning events on national scale are motivated through different attempts in the past in which governments had arranged to disconnect whole countries from the Internet, just to control traffic within that countries, as we elaborate in Chapter 2. In Chapter 7 we investigate the impact of isolation events in the underlay on routing mechanisms in the overlay. We discuss the spontaneous forming of communication islands during isolation events and present a solution to merge disrupted overlay parts again. For this purpose, we compare different solutions of overlay merging algorithms and propose our own solution, the Ring Reunion Algorithm [7], which alone is able to merge partitioned overlays reliably. We further show that our Ring Reunion algorithm reduces its activity during stable periods to reduce message costs.

### 1.3.2 The Assembly of Coexisting Overlays to an Overlay Stack

As we motivated in Section 1.1, a huge diversity of decentralized applications exists, out of which each overlay serves one specific purpose. We found out, that the parallel execution of different overlays on one peer has been examined rarely, although the subject of coexisting overlays has many open issues. One open point is to describe and characterize basic functionalities in current overlays and to find out how duplicated functionalities caused by multiple

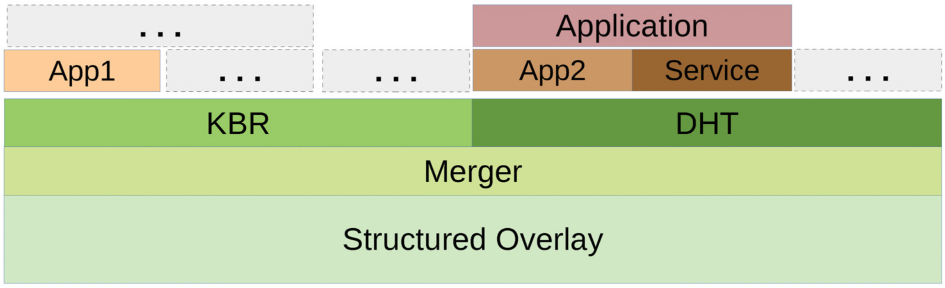


Figure 1.3: The overlay stack can be extended vertically, by adding new applications, services, or other modules on top of existing modules. Grey boxes with three dots indicate possible spaces for new modules.

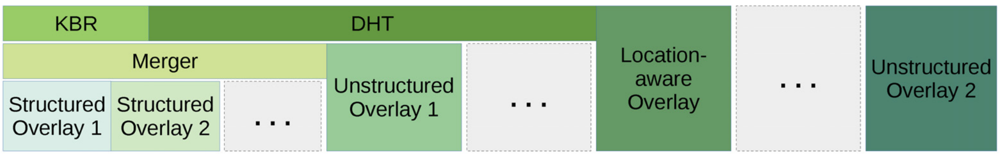


Figure 1.4: The overlay stack can be enhanced horizontally, by adding new overlays, or other modules into existing layers of the stack. Grey boxes with three dots indicate possible spaces for new modules.

overlays executed in parallel on a single peer could be avoided. In the following, we present our contributions to the issue of coexisting overlays given in this thesis.

The implementation and evaluation of contrasting P2P applications, which are motivated in Section 1.1, as contribution to this thesis is described in Section 1.3.1. Comparing the four P2P applications and the protocols behind them, we notice that a basic lookup service which provides routing solutions, for example given by a DHT, is necessary for the applications to be realized. A simple solution to reduce overhead produced by redundantly implemented functionalities is to use a single DHT as basis to provide a common lookup service for applications. Different applications are then easily added on top of this common lookup service. Advantage of this approach is that the underlying DHT can be exchanged and optimized separately from the applications above. Redundant mechanism for routing purpose are avoided as many applications share one DHT as basis. Furthermore, most DHTs already provide routing in $O(\log(N))$ to all participating nodes. Depending on the DHT used, it already provides mechanisms to keep the routing overlay stable in presence of churn. In Chapters 4, 5, 6, and 7 we present and evaluate diverse P2P applications built on top of a DHT with the goal that they can be operated on a common DHT, so that duplicate costs for maintaining a lookup service is avoided and new applications could be added additionally without effort. In Arsham Sabbaghi Asl's master's thesis [16], we operated all overlays presented in Chapters 4, 5, 6, and 7 on a

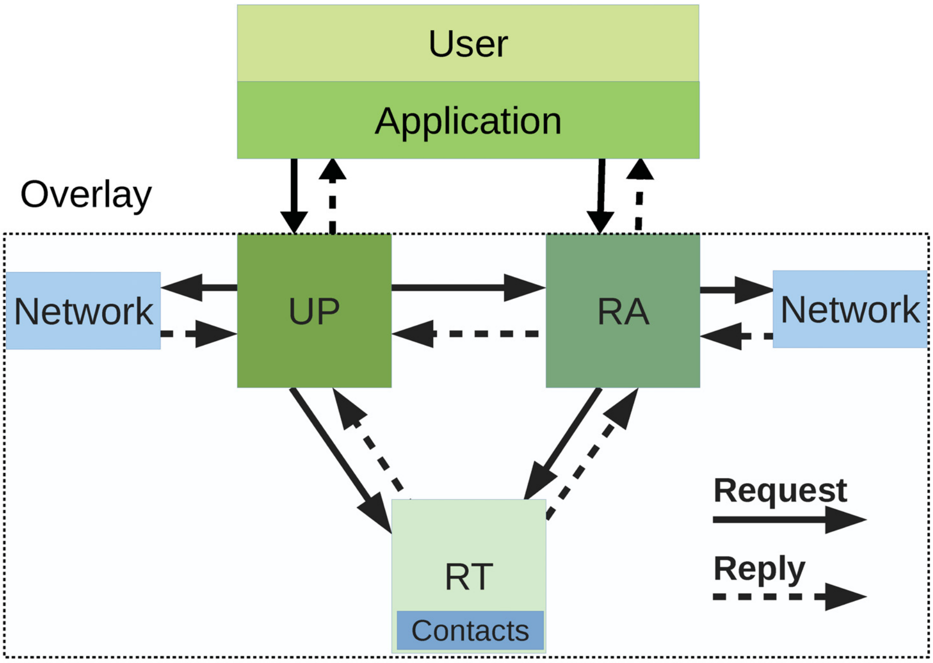


Figure 1.5: Overlays consist of three modules at least: a routing table, a routing algorithm, and update mechanisms. The join procedure is part of an update mechanism which can use the routing algorithm to find first contact nodes during the bootstrapping phase.

common DHT to analyze the behavior of DHT-based applications in our overlay stack. While it is clear, that a common DHT reduces overhead related to duplicate routing mechanisms, it can also be seen that coexisting applications that are coupled through a common DHT can influence each other massively. Depending on the applications stacked upon the DHT, and depending on the DHT itself, bad behavior affects coexisting applications as well as good behavior does.

Using a common DHT as underlying lookup service for different specific applications, we notice that a DHT alone might not be suited as basis for all applications. A question that arises is what if more routing overlays are needed, for example to provide a greater variety of routing functionalities? Especially, overlays for anonymous communication should not be built upon normal DHTs, in order to expose as less information about other participants as possible. Another reason, why one DHT is not enough as basis could be that a more specialized routing algorithm is needed for tasks which are not supported by normal DHTs.

We present in this thesis another approach of combining multiple overlays to an overlay stack which is independent from an underlying DHT. In this way, new routing overlays or lookup services can be added to the overlay stack, even if they are not DHT-based or belong to the

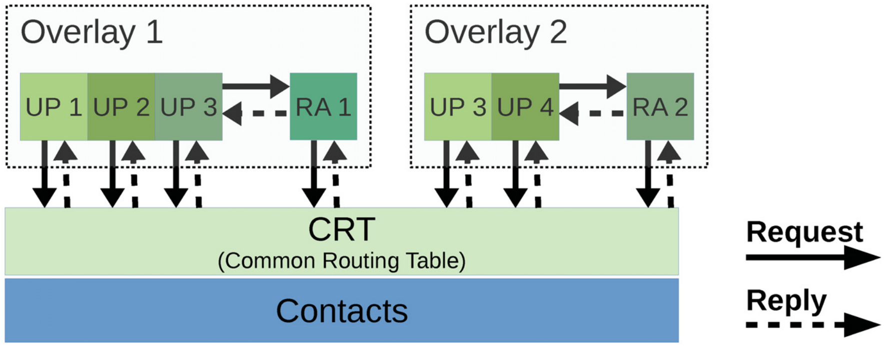


Figure 1.6: A common routing table can serve multiple overlays at once.

four classifications of applications presented in Section 1.1. Investigating the structures of multiple overlays and characterizing typical elements overlays are made of, we learn how the overlay stack as drafted in Figure 1.1 can be enhanced with further routing algorithms and other modules. Figure 1.3 shows that our overlay stack can be vertically extended by adding new applications on top of a common DHT, on top of any other routing overlay, or on top of existing applications. Instead, Figure 1.4 shows that the overlay stack can grow horizontally through the addition of new routing protocols.

In Chapter 8 and paper [9], we characterize and classify modules, out of which overlays are typically made. Additionally we summarize similarities between different overlay modules of the same type and we show differences to other types of modules. We show that overlays are basically made of three modules, namely a routing table ($RT$) in which contact data of other participants is stored, a routing algorithm ($RA$) which describes rules on how to forward messages, and optional update mechanisms ($UP$) which keep the overlay stable under churn. Figure 1.5 illustrates how the three overlay modules can be connected. It can be seen that the routing table is a passive module, which is accessed by other modules, but never requests data from other modules. Applications which provide functionalities apart from the basic overlay can be implemented on top of these modules.

The separation of overlays into at least the three proposed modules allows to optimize specific parts of an overlay. We investigate the characteristics of the different modules and describe their function in an overlay in our Chapter 8. We further contribute to the knowledge of stacked overlays by identifying parts in the overlay stack that can be optimized to reduce additional costs caused through the stacking of modules. In this way, we find out that most routing algorithms are too specific to have common elements. Moreover, each routing algorithm individually shapes the behavior of an overlay. The majority of routing tables instead, can be replaced by a common routing table for all coexisting overlays. Some contact data in a common routing table belonging to one overlay can be reused by other overlay modules, for example if the same identifier space and distance metric is used in two overlays, like in Chord and Pastry. Figure 1.6 shows an example of how different overlay modules, mostly routing algorithms are connected to a common routing table. In summary, we can say that a common routing table is related to other overlay modules like a common DHT is related to parallel applications.

In our paper [9] and in Chapter 8 we introduce a *common routing table* that replaces the routing tables of single overlay solutions in our overlay stack. We name different modules out of which overlays are made and characterize typical overlay algorithms. In our evaluation, we show for three contrasting overlays that their implementations in combination with a common routing table are not worse than the original protocols. We further show, that different costs regarding network traffic and stored contacts can be reduced through the use of a common routing table.

## 1.4 Outline

In this chapter, we introduce and motivate overlay stacks as well as related research questions. In the next chapters, we present our contributions which lead to the understanding of the overlay stack. This thesis is structured as follows:

Chapter 2 concludes our contribution on the impact of the Arab Spring on online social communication and infrastructure. We summarize the events during the Arab Spring in Egypt in 2011 which lead to a deletion of the BGP connecting Egypt to the rest of the world. We explain current mechanisms which are actively used by governments to overhear, censor or block Internet traffic. Additionally, we present techniques to escape those technical restrictions and we discuss the impact of the technical constraints during the Arab Spring on today's communication in online social networks and other Internet services. This chapter motivates the need for fully distributed anonymous and social communication platforms which are able to withstand strong network dynamics, including complete network partitioning.

In Chapter 3 we motivate the occurrence of P2P Networks, Mobile Ad-Hoc, and Opportunistic Networks and present current simulation tools for the different categories of networks. We compare different state of the art simulators for the three classes of networks and identify one simulation tool which allows evaluation of the most networks with least effort to change the tool.

Chapters 4 to 7 comprise solutions for the four contrasting cases, namely social-centric overlays with free communication (SF), social-centric overlays with restricted communication (SR), location-centric overlays with free communication (LF), and location-centric overlays with restricted communication (LR). The results of the evaluations of the four solutions can be found in our papers [72], [11], [8], and [7].

DiDuSoNet, our solution of a distributed online social network is presented in Chapter 4. In DiDuSoNet, each user owns a private web page, called profile, which can be filled with any social content like posts, pictures, comments, and other. In DiDuSoNet, profiles are copied to friendly nodes, the point of storage ($PoS$), in order to keep content available if peers leave the network A basic DHT like Chord or Pastry provides lookup functionality to find the PoSs of a specific profile. In our solution, we use the concept of Dunbar's ego networks, which enhance existing DHT with social-relationship topologies which base on social interactions of peers.

In Chapter 5, we present FRoDO, an overlay to provide anonymous communication achieved through routing over Dunbar-based relationships. We incorporate social Dunbar contacts into existing overlays and show that this naive approach is problematic in terms of lookup reliability. A solution to navigate through Dunbar contacts is presented with goLLuM. Advantage of our approach is its independence from the underlying DHT.

We present LobSter, our solution of a location-aware peer-to-peer overlay in Chapter 6. We compare our solution which uses space-filling curves to map two-dimensional data to a one-dimensional Chord identifier space with Geodemlia [70], a specialized approach in the field of location-aware overlays. Other than in related work, we focus in our evaluation on moving peers.

Although churn is an important topic to existing overlay networks, only few solutions consider the underlying network to fail. In Chapter 7, we focus on partitioning-aware overlays which are designed to merge separated overlay parts after network events that lead to isolated communication islands in the underlay. We propose our the solution, the Ring Reunion Algorithm which is able to merge Chord-like overlays reliably under realistic conditions.

We conclude our insights about the stacking of overlays in Chapter 8. We identify typical overlay functionality and requirements of structured and unstructured overlays and summarize them in one paper. With this knowledge at hand, we are able to build various overlay applications with a core set of functions.

Finally, we conclude our contributions and findings about stacked overlays in Chapter 9 and discuss possible future work in the field of stacked overlays.

# Chapter 2

# Webs of Change? The Transformation of Online Social Networks and Communication Infrastructures from a Technological Point of View

This chapter summarizes the contributions and gives a verbatim copy of our paper [10].

Kalman Graffi and Tobias Amft. "Webs of Change? The Transformation of Online Social Networks and Communication Infrastructures from a Technological Point of View". In: *Academia in Transformation, Arab-German Young Academy of Sciences and Humanities (AGYA)*. 2016.

In this article, we review the technology used to communicate and discuss their weaknesses which had allowed the government to interrupt those channels. Further, we present the current state of possibilities to combat the constraints in todays technology of communication networks, and we discuss the impact of the Arab Spring on research in the field of computer science. The political situations and uprisings during the Arab Spring presented here demonstrate the need for fully distributed communication platforms which survive targeted measures to block centralized infrastructures. Using peer-to-peer technology, communication between peers is less traceable and preventable in comparison to traditional server-based infrastructures. Chapters 4, 5, 6, 7 and 8 comprise different peer-to-peer applications and technologies which support fully distributed communication and reduce the possibility to censor Internet traffic as it has happened during the Arab Spring in Northern African countries, as described in this chapter and in our paper [10].

## 2.1 Paper Summary

The Internet has grown over many decades now and consists of billions of interconnected devices. Different protocols specify rules to exchange information between two remote devices.

The Internet comprises different parts, the autonomous systems, which are controlled by distinct authorities and interconnected through exterior gateway protocols. Connection to the globally accessible Internet is granted by an Internet service provider which manages the participation of users in an autonomous system. Technical possibilities to attack users are given a lot. We differentiate between active and passive attacks, whereas active attacks belong to the type of attacks which are able to manipulate traffic. Passive attacks are those in which data is only read but not changed.

Since forwarded data packets have to traverse many instances before they reach their destination, multiple (sometimes very easy) attacks can be used to block information delivery. Search engines for example are services on the Internet which allow to search multiple web sites for any search terms. A summary of the search is usually presented in form of another website. Depending on the country in which a specific search engine is used, it might hide and keep results so that only parts of the matching sites are presented to the user.

Deep packet inspection techniques allow to overhear unencrypted messages and manipulate them. In China for example, connections which include suspicious keywords are forced to be closed by local Internet service provides without knowledge of the user. The most drastic example of censorship was given during the Arab uprisings in 2011. Feared from the political situation and massive usage of online social networks, the Egyptian government deleted routing tables in gateway routers which connect the Egyptian autonomous systems with other parts of the Internet. On January 25, 2011, the Egyptian government instructed local Internet service providers to shut down their services. This decision was a reaction to ongoing uprisings during protests driven by the Arab Spring. As a result, more than 80 million inhabitants fell into a communication hole.

Different techniques can be used nowadays to overcome the described limitations. Encryption allows to hide content from any unintentional reader and allows private communication. Proxy routing is a technique to hide the real originator of a message inside a network. To achieve this, messages are forwarded to one or more participants (proxies) which carry a message further to the intended destination. A server which is contacted by a proxy is not able to identify the real originator of a request. In this way, country specific blocking of web sites can be circumvented. As a next step, a service could be implemented in a distributed fashion to leave out central server elements. Whereas traditional client-server-based services rely on a server as single point of service provisioning, P2P networks distribute user related data and so the control over it among participants. Through direct connection of communication devices (for example via Bluetooth, WiFi Direct, or USB stick), untrackable networks are formed, which are sometimes called darknets. During the last years, tools for private and anonymous communication, e.g. realized with multi-hop relaying and additional encryption, have emerged and accepted with great pleasure.

In an interview we found out that most of the existing tools for anonymous and private communication are difficult to set up and use for non technologists. We observe that the Arab Spring as well as the NSA affair (2013) in which Edward Snowden revealed the PRISM program used by the NSA for surveillance have not influenced the field of academic computer science exceedingly. In summary, the awareness of national monitoring programs has not led to the invention of significantly new security, anonymity, or privacy preserving protocols or techniques. Oppositely, the communication tools used during the Arab Spring seem to have acted like a catalyst so that an increasing number of people finds pleasure in its use. Moreover, recent events have shown that current technology which exists for many years now is still not fully understood by its users, governments and authorities included. As a result, the technological abilities to encrypt messages and to distribute them anonymously force different governments and authorities to discuss the general prohibition of cryptographic methods to hide information. We believe that through the ongoing growth of the Internet, many academic disciplines might form which are interested in the understanding of modern Internet services and impact of social-media platforms (and underlying technologies) on its users.

## 2.2 Contribution

Our contribution in this highly interdisciplinary article is given as follows:

- We analyze the events which lead to a communication breakdown in Egypt during uprisings of the Arab Spring. We identified that deletions of routing entries in the border gateway protocol have led to a separation of the Egyptian network from the rest of the world. Additionally, we roughly describe the structures which in combination represent today's Internet in a way that it is understandable to non computer scientists.

- We review current technology for online social communication and summarize common techniques used by attackers (mostly oppressive regimes in our examples) to censor or even manipulate Internet traffic. On the contrary, we also describe current techniques to circumvent the presented attacks.

- Finally, we discuss possible changes in academic thinking caused by recent events like the Arab Spring (2010-2012), the NSA affair (2013), and other attempts to censor Internet traffic.

The article is mainly written to explain current technology to people which are not familiar with academic computer science or related, technology-oriented topics. Our goal is to give a first insight into the topic of security, privacy, and anonymity to the interested reader. Additionally, with this article we strongly motivate the advantages of distributed communication protocols under modern developments of the Internet.

## 2.3 Personal Contribution

Tobias Amft, the author of this thesis, contributed to this paper (book-chapter) by investigating historical and political events around the Arab Spring. With this knowledge at hand, he wrote major parts of this article. He further reviewed state of the art tools for private and anonymous communication. Finally, he contributed with a discussion about the possible impact of the Arab Spring on academic thinking in computer science. Kalman Graffi motivated and started this article. He was involved in different discussions regarding shape and content of the article. Kalman Graffi proofread the article several times and provided valuable information about different peer-to-peer techniques which are reviewed in this article.

## 2.4 Importance and Impact on Thesis

The paper gives a general insight in problems in modern Internet society which is strongly influenced by censorship. We compare classic client-server approaches for social networks with emerging distribute peer-to-peer alternatives for private and anonymous communication. The article summarized in this chapter illustrates the need for anonymous and fully distributed communication networks which complicate the blockage or censorship of communication platforms. We highlight the main advantage of peer-to-peer-based communication tools over server-based platforms which is to distribute the control about data among its users instead of being bound to a single operator. Furthermore, distributed networks are more robust against censorship or surveillance, for example, conducted by oppressive regimes or authorities. In the next chapters, we present different peer-to-peer applications and techniques which foster the here (in this chapter and during the Arab Spring) desired solutions for distributed communication and distributed data storage.

# Chapter 3

# The State of Simulation Tools for P2P Networks on Mobile Ad-Hoc and Opportunistic Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [32].

Ahmad Cheraghi, Tobias Amft, Salem Sati, Philipp Hagemeister, and Kalman Graffi. "The State of Simulation Tools for P2P Networks on Mobile Ad-Hoc and Opportunistic Networks". In: *Proceedings of the International Conference on Computer Communication and Networks (ICCCN)*. 2016. Acceptance Rate: 30%

In some countries of the world, communication between individuals is still suppressed by the local government. Since discussions and communication as well as big parts of social life are shifted into online social networks and other services running on the Internet, the respective governments do not hesitate to block or censor Internet traffic in their country. Especially services and applications of which servers are located outside the respective country become subject to the blocking of Internet traffic. Different approaches for communication services emerge out of a necessity in the last years. Most promising are distributed approaches of networks which are not dependent on a centralized server that could be blocked. In our paper [32], we focus on a possible combination of peer-to-peer (*P2P*) networks, opportunistic networks (*OppNets*), and mobile ad-hoc networks (*MANETs*), which are three different network types being promising censor-resistant alternatives to centralized server approaches.

Different evaluation methods exist to rate new protocols in the three categories of networks. Analytical evaluations in which problems and protocols are formulated in mathematical expressions, simulation-supported evaluations in which the basic behavior of solutions is implemented and tested in a simulated environment, or testbed-driven evaluations in which real devices and real-world scenarios are analyzed, allow us to investigate the behavior of different protocols under specific circumstances and test conditions. The advantage of analytical evaluations is the provable correctness of possible solutions. Although those analytical evaluations are cheap in terms of equipment, the disadvantage lies in the high costs in terms of complexity. Mathematical expressions often limit evaluations to basic evaluations under the cover of well-chosen assumptions for the environment in which the analysis takes place. Real-world testing environments promise to produce best results in terms of accuracy. The downside of testbed-driven evaluations is explained by the equipment costs and set up complexities of tested scenarios. Simulation environments allow to combine the benefits of the two other approaches. On the one hand, simulations require not more than a simulation software and a computer to analyze given protocols. The complexity of the simulations can be adjusted reaching from basic descriptions of a protocol to almost real-world behavior of a solution.

The goal in our paper [32] is to identify suitable simulation environments which cover P2P, OppNets and MANETs as good as possible.

## 3.1 Paper Summary

Internet traffic is censored or recorded in many countries in the world. Centralized server approaches are prone to well aimed attacks on the infrastructure of the corresponding services. Distributed approaches realized with peer-to-peer networks, mobile ad-hoc networks, or opportunistic networks are less interruptible since the underlying infrastructure is provided and set up by its users. Especially online social networks and communication software could benefit from a distributed infrastructure, since user data is not maintained/gathered at one specific point in the Internet. In order to take down a client-server based solution, simply the few servers providing the platform need to be shut off. On the contrary, in order to shut down P2P networks, OppNets, or MANETs, each single node in the network have to be shut down, which is infeasible.

Peer-to-peer networks on the one hand allow users to alleviate the drawbacks of central server approaches. Through the use of distributed applications that implement both, client and server part of a necessary infrastructure, participants are difficult to be identified personally by an observer. Additionally there is no central server that could be blocked for connections withing the respective country. Whereas peer-to-peer networks usually utilize the Internet as underlay, mobile ad-hoc networks (*MANET*s) provide their own network infrastructure. Participants in MANETs are interconnected, mobile devices which are equipped with radio interfaces for the purpose of wireless communication and free to move arbitrarily in any direction. Mobile ad-hoc networks (*MANETs*) allow data packets to be forwarded between local neighbors until a destination is reached. In this way, no dependencies in an underlay arise, since devices communicate directly without any foreign link in between. If no path between origin and target exists, messages are dropped, since nodes do not store forwarded messages. Opportunistic networks on the other hand, which are subclass of delay tolerant networks (*DTN*s) are aware of long propagation delays which would lead to failures in MANET protocols. Oppositely to MANETs, nodes in opportunistic networks store messages until a suitable contact is found to be next hop of the message. Opportunistic networks use opportunistically created connections to forward data towards a target, even if an end-to-end path may never exist. A store, carry and forward mechanism is used in opportunistic networks to route data from any initiator node to a desired target, even in presence of long delays. Doing so, connections between participants are spontaneously created if they have the opportunity to exchange the data, e.g. while they meet at one location.

The combination of P2P, MANET and OppNets could be done as follows: a MANET layer provides basic communication based on wireless technologies like WiFi, or Bluetooth. An

OppNet on top provides long latency communication and handles forwarding of data between communication islands. A peer-to-peer layer on top provides logical routing structures, lookup functionality and distributed storage abstraction. Finally, different applications on top of the protocol stack solve different requirements of the user. A combination of the protocols could enable efficient file sharing between mobile devices in remote locations which are absent from any connection to the Internet. P2P solutions could coordinate file indexing, using an underlying MANET as underlay. Another case could be to spread data of interest through the combination of P2P file sharing approaches on top of an opportunistic network. Files could be exchanged between bypassing participants until data reaches a desired destination. During civil uprisings, or catastrophes, a combination of all approaches could be used to provide a reliable communication tool.

Simulators are well suited to test large scale protocols used in P2P, MANET, and OppNet systems. Within our paper [32] and this chapter, we summarize the current state of simulation tools for P2P, OppNets, and MANETs. Our aim is to identify appropriate simulators for the three topics on the one hand. On the other hand, we try to identify which simulator environment is best suited for combined evaluations of P2P, OppNet, and MANETs at the same time. A multi-purpose simulator which supports these three types of networks has to fulfill multiple requirements. It should be actively maintained, e.g. by a community, so that help is available. A simulator should (for our purpose) be event-based so that smallest interactions can be investigated in detail, it should be scalable to support the simulation of thousands of users. Finally, a simulator should support multiple layers in order that different types of systems can be implemented. In the category of P2P simulators, we compared 9 existing tools, finding that PeerfactSim.KOM [61] [108] and Oversim [18] [105] are best suited to realize the combination of different network types. For the category of MANETs, we compared 12 simulators out of which OMNeT++ [140] [104] in combination with InetMANET and NS-3 [74] [103] are a good choice. The ONE simulator [80] [137] is the only active simulator for OppNets.

In conclusion, all compared simulators need to be modified to support all three types of networks (P2P, MANET, OppNet). Nevertheless, PeerfactSim.KOM and NS-3 seem to provide most functionality for the combination of P2P, MANETs, and OppNets.

## 3.2 Contribution

The contributions in this paper to the thesis are the following. First, we introduce in this paper the characteristics of P2P networks, OppNets, and MANETs describe their differences. As a next step, we propose different use cases which require either a full or at least a partial combination of the networks. Second, we list functional and non-functional requirements for a simulation environment which allows to investigate a combination of peer-to-peer, MANET, and opportunistic network protocols. We review state of the art simulators typically designed for the three networks and categorize them with respect to the requirements to the three single network types. For each type of network, the related routing layer is characterized and out of that, a suitable simulation environment for each network type is identified. Finally, we conclude our research and identify a simulation environment which is, in our opinion, best suited to support simulations of combined P2P, MANET, and OppNet protocols.

## 3.3 Personal Contribution

The reviewing and selection process of the IEEE International Conference on Computer Communication and Networks (ICCCN) 2016 resulted in an acceptance rate of 30%. Ahmad Cheraghi analyzed existing work on MANET simulators, he conducted the paper submission, and organized the final version of this paper. Tobias Amft, author of this thesis, collected and summarized information about existing peer-to-peer simulators. Furthermore, he organized and shaped a first version of the paper. Salem Sati reviewed OppNet simulators and summarized them. Philipp Hagemeister motivated the application case of the paper. Kalman Graffi contributed to the motivation and the methodology of this paper and guided the production of it.

## 3.4 Importance and Impact on Thesis

The paper summarized in this chapter supports this thesis in different ways. First, the paper motivates interesting use cases regarding distributed infrastructures like P2P networks. Similar to the previous Chapter 2, this chapter demonstrates the needs of distributed communication network, especially in countries in which communication is occasionally or permanently blocked or surveilled by the government or any oppressive regime. Second, in this chapter, we present a survey on the current status of simulation tools for P2P networks, OppNets an MANETs. We list simulation tools proposed for the three types of networks and focus on different criteria to identify the best suited simulator environment for the respective purposes with the further vision to simulate all network types in one simulator. Moreover, through the systematical evaluation and in-depth investigation of available simulation environments for P2P networks, we are able to identify an appropriate simulation environment used in this thesis to evaluate the variety of P2P protocols, algorithms, and techniques described in the following Chapters 4, 5, 6, 7 and 8. We identified PeerfactSim.KOM to be the best choice for the simulation of P2P networks, because it is designed to conduct experiments with P2P networks, it is actively maintained, and we stay in close contact to further active and former developers.

# Chapter 4

# DiDuSoNet: A P2P Architecture for Distributed Dunbar-based Social Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [72].

Barbara Guidi, Tobias Amft, Andrea De Salve, Kalman Graffi and Laura Ricci. "DiDuSoNet: A P2P architecture for Distributed Dunbar Based Social Networks". In: *Springer Peer-to-Peer Networking and Applications, Special Issue of Springer Journal Peer-to-Peer Networking and Applications (PPNA)*. 2014.

Online social networks (OSNs) grew successfully in the last years and have changed the way how people interact socially. Typically, users store huge amounts of personal data in online social networks. Once users have published their data online, they lose control over them, as they are transferred to a remote server. In the last years, research follows the trend of distributed online social networks (DOSNs) which base in most cases on a peer-to-peer solution. Advantage is that no single service provider maintains user data and therefore no central point exists which could be compromised or shut down. The main challenge in distributed approaches is to keep data available in the network even if participants join and leave it frequently. Existing solutions try to preserve data availability by selecting friends in the network which take over the responsibility for a node's data in case it is absent. Another challenge in these networks is first to find all sets of replicated data and second to identify the last recent version of data.

## 4.1 Paper Summary

In this paper, we present *DiDuSoNet* (Distributed Dunbar-based Social Networks), which combines concepts from social sciences with state of the art in the field of distributed online social networks. DiDuSoNet consists mainly of two layers. It bases on a DHT which provides lookup functionality and solves the bootstrap problem which is the challenge to learn about other, sometimes friendly, nodes in the system when a peer is joining for the first time. On top, a social overlay is located which offers direct point-to-point communication between peers, information diffusion between many peers, as well as further social services. Each participant in DiDuSoNet is associated to a unique social identifier and owns a profile which is a personal web page that contains social content like texts, posts, comments, pictures, and many other user-related content. The user decides which content is available to the public and which data is private so that it can only be accessed by selected friends.

Robin Dunbar, a British evolutionary psychologist, observed that human cognitive capabilities to maintain social relationships are limited [55]. According to Dunbar, the human brain is able to associate approximately 150 active relationships only. The limit of 150 contacts is known as the Dunbar number and is further divided into four affinity groups, based on the interaction frequency of an *ego* with a regarding relationship. We adopt the findings of Dunbar and define *ego-networks* to be the construction of an *ego* which is connected with multiple contacts, the *alters*. The stability and strengths of the connection between an ego and each alter is measured through a numerical value, the *tie-strength*.

In order to increase the availability of content during churn, egos instruct selected friends to store replicas of their content. Doing so, a node selects a small subset of its friends with each friend being a point of storage (*PoS*) for the replicated data. The selection of suitable PoSs depends on the following points:

- **the tie-strength** defines the contact frequency between an ego and a friend.
- **the mean session length** defines the average duration of a user session.
- **the connection gain** defines the resulting benefit of gaining knowledge about other nodes through the connection to a friend. The gain is proportional to the number of common friends two nodes have.

As long as the original owner of content is online, he is also PoS for his own data. Information about how to contact the PoS is stored into the underlaying DHT. In this way, the DHT is used as lookup service which enables to find the PoS associated to a given social identifier. In order to obtain content of another user $v$, a user $u$ has to lookup the address of at least one PoS of $v$'s data in the DHT first. Thereafter, user $u$ contacts one of the PoSs and requests the content related to user $v$. Public key authentication is used to identify the access permissions user $u$ has on the content. In this way, peers are able to decide who is able to read or change specific parts of the personal profile. Through the PoS table maintained in the DHT, each user always knows the current PoS for specific content and can access the newest version of an online profile.

To increase robustness against single node failures, the different PoSs of one specific profile are connected. Each PoS observes the status of several alternative PoSs to enable quick reaction to node failures. If a PoS leaves the network, with or without notification, the remaining nodes select a new PoS out of the owner's friend set automatically. Previous PoSs or the owner of a profile contact the current PoSs if they join the network again so that they obtain the latest version of the profile. In case that all PoS fail, the first node which has previously been PoS and has stored the profile locally can offer this (maybe outdated) version of the profile until a PoS with a newer version comes online again. The number of PoS should be rather small to avoid costs for content replication and updates.

We simulated and evaluated DiDuSoNet with the *PeerfactSim.KOM* peer-to-peer protocol simulator[1]. From *SocialCircles!*[2] [47], we obtained real-world Facebook relationships from more than 600 users for our evaluation. In our simulations, each node publishes an own profile and enables the data availability protocol described above. Evaluation shows that in first tests, approximately 95 percent of all published profiles are available at any time in the network. In a scenario of 48 hours simulated time, we observe that after a time mostly 99 to 100 percent of all profiles are available with a maximum of only two PoSs per node. We further present the average tie-strength between egos and its PoSs in comparison to the tie-strengths between an ego and its other friends. Finally, we investigate the availability of profiles depending on the number of friends an ego maintains. We see that up to 4 friends are necessary to reach approximately 90 percent reachability.

## 4.2 Related Work on Social Overlays

Distributed online social networks (*DOSN*s) [44] allow their users to provide own infrastructures to maintain their data. The greatest benefit of DOSNs is that users keep control over their data as it is not transferred to a remote service provider. Main challenge in distributed online social networks is to guarantee data availability in a dynamic environment. This and further challenges in DOSNs are described in [43] and [107]. In general, existing solution can be divided in friend-based or DHT-based approaches.

This first category of solutions comprises overlays which are built on trusted or self-hosted infrastructures. Typically, friend-to-friend [26] relationships, trusted nodes, or private servers are considered to manage user data. Users in Diaspora[3][4] and FOAF [59] for example set up private servers. Safebook [39] or GoDisco [43] on the other hand leverage friendship topologies for building their infrastructure. Different approaches which consider friendly nodes as data storage are presented in [88], [138], and [130]. The authors of [95] show that availability of data hosted by friends is at risk assuming certain user behavior. Among the category of friend-based social networks, some approaches assume special friendship behavior. Dunbar observed ego networks [116] and introduced the Dunbar number [55] to classify human relationships. The presence of the Dunbar number has been verified in several online social networks [134, 14]. [36] verified the presence of Dunbar's number in online social networks like Facebook. A measure to rate friendship relationships is the tie strength which is introduced in [60, 13]. Further studies about the behavior of users in online social networks in order to define data allocation strategies is given by De Salve et al. in [45] [46]. Especially, the strucures of communities and their evolution over time in Facebook is presented in these works.

The observation of social behavior lead to different DOSNs which focus on Dunbar's observation [71], trusted services for data availability [38], or information diffusion [37]. Observations show, that only a subset of friends is really active [14, 13] and OSN behavior correlates to friends [24]. The authors of [22] and [96] predict users' availability based on their behavior. Some approaches propose different data placement strategies which are not dependent on friendship topologies. The authors of [101] and [122] propose different replica placement

[1]http://peerfact.com/
[2]https://www.facebook.com/SocialCircles-244719909045196/
[3]https://joindiaspora.com
[4]https://diasporafoundation.org/

techniques to guarantee data availability. In [106], the authors propose to distribute small data blocks instead of replicas among online nodes. Cachet [102] requires collaboration of participants to make data available.

Solutions in the second category of social overlays, namely DHT-based social overlays, build specific applications atop existing DHTs to benefit from existing structures. SROUT [93] for example investigates social links in Chord. LibreSocial[5], formerly named LifeSocial [62, 64, 65], is a multi-layer framework for distributed social networks which is based on FreePastry[6]. PeerSoN [27] (based on OpenDHT [115]), SuperNova [129], and OverSoc [143] followed the idea of LibreSocial. Since DHTs allow to access data publicly, secure distributed data structures [77], access control [67] and other features are needed.

## 4.3 Contribution

In this paper, we present a distributed online social network approach built on top of a DHT lookup service. Doing so, we solve two challenges for distributed online social networks. First, we solve the bootstrap problem through the use of a DHT lookup service which allows newly joined peers to find first contacts and friends in the network. Second, we propose a protocol to ensure availability of user content in a frequently changing environment. Novel in the field of distributed online social networks is the combination of Dunbar's research from psychologies with social peer-to-peer networks. We introduce the tie-strength, which is the strength of a relationship between two participants, as a measure to select replica nodes for a user's content. A user selects friends with a high tie-strength as points of storages to store personal data on them. We show that with the Dunbar approach it is sufficient to have two PoSs for one profile to keep it available.

## 4.4 Personal Contribution

Barbara Guidi motivated and organized the paper. She summarized related work, described the system model, introduced the idea of a Dunbar related DOSN with trusted social storage, and described the simulation methodology. Additionally, Barbara Guidi implemented the user-application part of the social overlay and concluded the paper. Tobias Amft, author of this thesis, took part in writing and shaping of the article. Amft suggested a DHT as basic lookup service which forms the basis for a data replication service. He implemented basic communication patterns for the user-application and he designed and implemented the data availability service described in *Part 5* of the article. Amft conducted experiments in PeerfactSim.KOM and contributed with the evaluation of the Dunbar related point of storage data-availability service summarized in *Part 8* of the paper. Andrea De Salve contributed with several writings and suggestions to the implemented application. Kalman Graffi and Laura Ricci supervised and reviewed the paper and made valuable suggestions and annotations regarding DOSNs in various discussions.

[5] http://libresocial.com
[6] http://www.freepastry.org/freepastry

## 4.5 Importance and Impact on Thesis

The presented paper is important to this thesis for a number of reasons since it allows to answer many of the research questions listed in Section 1.2. We show with this paper that it is possible to assure data availability up to a certain degree in Dunbar-based ego networks. The tie-strength as indicator of strong relationships helps to select appropriate friends as points of storages for personal user data. Moreover with our approach a multi-layer solution is presented, as we utilize a DHT as basic lookup service on which an application specific overlay is placed. Separating the DHT from the application overlay, we are able to exchange the underlying lookup service at any time. Additionally, we identify a lookup service with DHT functionality as possible basis for distributed online social networks.

As described in Chapter 1, we investigate the question if different applications can share a common DHT as basic lookup service. Such a common DHT reduces storage and networking costs in the way that only one DHT is needed instead of several DHTs, each being part of one application. A great benefit for other solutions presented in this thesis is that the underlaying DHT as basic lookup service allows newly joined peers to find contacts in the network. Applications using the same DHT which the presented distributed online social network is using, benefit from the existing routing table and are able to use already known contacts for bootstrapping. Similarly, we investigate in Chapter 8 the use of a common routing table which provides contacts to different overlays like a common DHT provides contacts to different applications on top of it.

# Chapter 5

# FRoDO: Friendly Routing over Dunbar-based Overlays

This chapter summarizes the contributions and gives a verbatim copy of our paper [11].

Tobias Amft, Barbara Guidi, Kalman Graffi, and Laura Ricci. "FRoDO: Friendly Routing over Dunbar-based Overlays". In: *Proceedings of the International Conference on Local Computer Networks (LCN)*. 2015. Acceptance Rate: 30.3%

Since users of online social networks (*OSN*s) loose control over their data, the distribution of online social network infractures has become an interesting topic in research. Although DOSNs provide good solutions to overcome the privacy issues related to OSNs, they introduce another critical problem, namely that participants in DHT-based DOSNs are visible to other peers and can potentially be contacted by every other participant in the network. In contrast, friend-to-friend (*F2F*) solutions reduce the risk of malicious behavior and foster the node's anonymity since peers are connected, according to real-world friendship relations, to trustful nodes only. The drawback of current F2F networks is that they are not structured and thus do not allow direct lookups for data. A combination of both worlds, having DHT-like structures with embedded friendship structures would allow to lookup data without peers being exposed to untrusted participants in the network.

In Chapter 4, we have proposed DiDuSoNet, a DOSN which bases on a DHT and exploits Dunbar-based ego networks to define friendship between peers. According to Dunbar, the human brain is only able to maintain approximately 150 social contacts actively. Characteristic of these ego-networks is that friendship is not necessarily defined bidirectionally. Thus it might happen that a peer $p$ maintains contact information to peer $q$ but not vice versa. It is not clear though if it is possible to route messages between arbitrary nodes in connected ego-networks which basically belong to the class of F2F networks.

In our paper [11], we show that naive modifications of existing DHTs to enable trusted routing is not possible. Instead, we identify requirements which have to be fulfilled by routing algorithms to be able to allow trustful lookups over friendly Dunbar relationships. As a result, we show how social-relationship graphs can be embedded into existing DHTs and we investigate how many friendly contacts are necessary in our *social DHT* to enable structured routing via Dunbar friends.

## 5.1 Paper Summary

Routing functionality which forwards messages via friendly nodes only has many benefits. Most of all, an overlay realized through F2F connections enables anonymous communication since messages are exposed to trusted nodes only. The use of a DHT for DOSNs is critical in terms of anonymity, since contact information is exposed to any participant.

In several studies about social relationships, the presence of Dunbar's circles [55] has been observed to reflect social ties. The circles around an ego node characterize the level of trust an ego gives to its friends. Up to 5 closest contacts are part of the *support clique*. As second level, the *sympathy group* concludes up to 15 friends, including the support clique. The third level of intimacy is called the *affinity group* and involves 50 contacts. Level four comprises all 150 active friends and is therefore called the *active network*. To support different levels of trust in an overlay, we integrated the four levels of contacts into the structured overlay Pastry [118]. The idea is to use only 5, 15, 50, or 150 contacts in Pastry for routing. In case that a user has too few friends, one could also use original Pastry contacts for routing which are, oppositely to friendly nodes, not trusted. In this way users can choose between different levels of trust for routing. With this approach that we call *SocialPastry*, we observed that this naive approach suffers either from little number of routing contacts or from resulting routing loops that occur frequently.

The greatest problem of embedding social-relationship graphs into DHTs and routing over Dunbar friends is that an ego's vision on the network is limited to its friends only. The friends of friends remain therefore unknown and can not be considered for routing and forwarding of messages. In this way it is not clear in the naive approach if messages are forwarded in target direction or in opposite direction with every hop. To enable routing over one-directional contacts, we propose an approach called *"going over Local Links using Dunbar's Method"*, in short *goLLuM*, a depth-first search variant which operates on top of existing DHTs. In this solution, an arbitrary underlying DHT like Chord [133] or Pastry is used to give every peer a personal identifier and to determine the responsibility range of a peer in the identifier space. Our goLLuM algorithm uses the knowledge about identifiers of friendly contacts, to select next hops according to their closeness to a given target identifier. In case that a node has no friends in its contact list which are closer to the target, the forwarded message is routed back to the last sender, with the goal, to choose another next hop towards the target. In this way, obstacles in the routing path and routing loops can be detected and avoided.

For our evaluation, we simulated the behavior of the goLLuM algorithm with different models of friendship distribution. We compared Small World graphs [141] to scale free graphs [17] which follow a power-law distribution and two real world measurements taken from Facebook[1]. According to the Dunbar circles, nodes categorize their friends into different levels: up to 5 friends are assigned to level 1, the next 10 friends are assigned to level 2, further 35 are in level 3, up to 100 nodes are associated with level 4. All further known contacts are treated like normal, not trusted overlay nodes. A more precise explanation of the goLLuM algorithm and SocialPastry, as well as the results of our studies, can be found in our paper [11].

In our simulations, we restricted the naive approach SocialPastry and the goLLuM algorithm

[1] http://current.cs.ucsb.edu/facebook/

to forward messages only via friends up to a certain level. The naive approach suffers to deliver messages more and more, the less friends for communication exist. It can be observed, using goLLuM, that routing via friendly nodes is possible even if nodes have few friends only and friendship is assigned one-directionally. On the other hand, if no path between two nodes exist, the sender of a message receives a notification that no path exists.

## 5.2 Related Work

Anonymity and privacy are two distinct characteristics of communication which can be explained the following: anonymity allows to hide sender, receiver, or both participants of a communication channel. Privacy on the other hand hides the matter of the communication from any other person but the communication partners. RetroShare [2], OneSwarm [3], WASTE [4], and Galet [5] are only few examples for anonymous communication tools. Distributed solutions for anonymous and private communication can be divided into the categories F2F [26] or private server approaches, group-based channels, or publicly available anonymous solutions.

In the first category, participants communicate via trusted connections (friends) or set up their own communication environment. FOAF [59] and Diaspora[6][7] assume users to setup private servers. Hamachi [8], Turtle [112] and GNUnet [20] assume secure communication through virtual private networks (*VPN*). FreeNet [120] since version 0.7 ([34]) assumes participants to be connected according to the small world phenomenon [141]. Communication in this category may additionally be encrypted. In the second category, messages are typically encrypted with a group specific key. Groove [139], Alliance [9], Direct Connect [10], OnShare [11], and Hybrid Share[12] operate this way. Also private server approaches may secure communication with a common private group key. The third category of anonymous and private communication approaches typically uses multi-hop relaying to hide communication paths, first introduced with Chaum's mixes [30]. Anonymous peer-to-peer file sharing applications are FreeNet [56], GNUnet [20], ANts[13], Kerjodando [14] (an extension of ANts), RShare[15], Mute[16], Share[17] and Winny[18]. Self-Chord [58] is a bio-inspired, ant-based approach, which uses concepts of ant colony optimization [53] to hide sender and receiver from each other.

## 5.3 Contribution

Our contribution in this paper is goLLuM, a depth-first search variant which forwards messages towards target nodes via friendly nodes. To do so, an underlying DHT is utilized to obtain information about the responsibility range of contact nodes. During routing, friends which are

[2] http://retroshare.sourceforge.net/
[3] http://oneswarm.cs.washington.edu/
[4] http://waste.sourceforge.net/
[5] http://galet.sourceforge.net/
[6] https://joindiaspora.com
[7] https://diasporafoundation.org/
[8] https://secure.logmein.com/products/hamachi/
[9] http://www.alliancep2p.com/
[10] http://dcplusplus.sourceforge.net/
[11] http://www.onshare.com/
[12] http://hybrid-share.sourceforge.net/
[13] http://antsp2p.sourceforge.net/
[14] http://www.kerjodando.com/
[15] http://rshare.de/
[16] http://mute-net.sourceforge.net/
[17] http://www.uguu.org/share/
[18] http://winny.cool.ne.jp/

possible next hops are sorted according to their distances to the target identifier. In this way, goLLuM forwards messages to any target node in a semi-structured way.

We show for different friendship models that goLLuM is able to find a route between two peers, if one exists. If no route exists, the algorithm informs the user about the absence of a path. Since messages are only forwarded between friendly nodes, our approach can be used to apply anonymous communication. In contrast to other approaches which assume mutual friendship (if peer $p$ trusts peer $q$, peer $q$ trusts also peer $p$), we show with our approach that routing is also possible if trust is defined for one direction (peer $p$ might trust peer $q$ but not vice versa).

Further more, we show that DHTs can be a basis for unstructured F2F-networks, thus allowing also friend-less users to participate in the network which is beneficial for a realistic overlay with joining nodes. Additionally, we propose a way to embed a social-relationship graph into a DHT and to route using trusted links. In more details, we investigate how many friends are needed for trust-supported routing in a DHT. Both approaches, F2F-routing and DHT routing come with dedicated benefits such as trust and security on the one side and general applicability on the other side. By combining these two worlds, we enhance reliable DHT routing with the help of trusted nodes. Furthermore, we define a stronger notion of trust based on the strength of a relationship defined through the Dunbar concept.

## 5.4 Personal Contribution

The reviewing and selection process of the 40th IEEE International Conference on Local Computer Networks (LCN) 2015 resulted in an acceptance rate of 30.3%. Tobias Amft, who is author of this thesis, organized and wrote major parts of the paper, he initiated research on Dunbar-based routing approaches. He designed, implemented and evaluated protocols described in the paper. Barbara Guidi reviewed the paper several times, added writings and restructured the paper. She motivated the Dunbar-based overlay and participated in discussions that lead to the goLLuM algorithm. Kalman Graffi and Laura Ricci additionally reviewed the paper, they took part in frequent discussions about the solution and the evaluation methodology.

## 5.5 Importance and Impact on Thesis

This paper is an answer to the research question on how to realize anonymous overlays with the Dunbar concept. We use a DHT as underlying basis lookup service and implement a lightweight solution for the specific application which enables friendly routing, i.e. routing over friends solely, on top. We identify with this work that it is a necessary functionality of a DHT to provide knowledge about a peer's responsibility range to all higher overlays. In this way, overlays and applications on top of the DHT can process this knowledge about responsibilities to provide further services.

Moreover, we show with our paper [11] that existing DHTs can be enhanced with social-relationship graphs so that anonymity and trusted routing via trusted contacts can be provided. Different applications sharing a common DHT benefit from the trusted and anonymous routing in the way that data is spread over friendly nodes only. A plain DHT in the overlay stack, introduced in Chapter 1, can be extended with the goLLuM algorithm so that it turns into a *social DHT*, without the need to change it. Other applications as described in Chapters 4, 6, and 7 can use the same social DHT provided by the overlay stack for their needs. New applications can be built atop the existing DHT or on top of other routing algorithms which share a common routing table and are part of the overlay stack, as described in Chapter 1.

# Chapter 6

# Moving Peers in Distributed, Location-based Peer-to-Peer Overlays

This chapter summarizes the contributions and gives a verbatim copy of our paper [8].

Kalman Graffi and Tobias Amft. "Moving Peers in Distributed, Location-based Peer-to-Peer Overlays". In: *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*. 2017. Acceptance Rate: 29%

Many peer-to-peer overlays have been realized in the last two decades. Most online services have the potential to be realized as distributed peer-to-peer applications, location-aware applications to be taken for example. Whereas computers were rather fixed to one location when the first peer-to-peer protocols came up, computers are small devices nowadays that can be carried to and used in any place. Location-aware applications which associate peers to physical locations and allow spatial searches are an interesting research topic again, especially if peers are changing their location frequently. The challenge of distributed location-aware systems is to find a suitable way to organize participating peers in order to achieve good routing behavior in terms of lookup success and hop minimization.

## 6.1 Paper Summary

In this paper, we present *LobSter* (*a Location-based System over DHT*), constituting a simple distributed location-based application which is built on top of Chord but can potentially be executed on top of any DHT overlay. LobSter uses space-filling-curves to map two dimensional data to one dimension that can be maintained in the underlying DHT. Contrary to specialized overlays in the field of location-aware solutions, our approach is loosely coupled to the underlying DHT, so that both, our application, or the DHT, can be optimized or exchanged without influencing each other. We evaluate our solution in a nearly realistic environment with moving peers. Additionally, we compare LobSter to an existing dedicated overlay, namely Geodemlia [70], which is designed without intention to separate the application from the basic overlay functions, to operate area searches.

We use z-order curves in our solution to map multi-dimensional points to an one-dimensional identifier. Using this mapping technique, the bits of the x-coordinate and the bits of the y-coordinate of a peer are bitwise interleaved. As a result, the one-dimensional z-key is obtained and assigned to a peer. In this way, each peer can be associated with an one-dimensional identifier without losing information about the peer's physical location. At every moment we are able to calculate a z-key according to a peer's position as well as the peer's position out of the z-key. In this way, we are able to sort peers linearly according to their spatial position.

To provide a spatial lookup service, each peer periodically calculates a z-key according to their current position. The resulting value as well as the contact information of a peer are then stored into the underlying DHT in which all peers are participating. Each node in the DHT stores now contact information to peers whose z-keys fall into the DHT node's responsibility range. Conversely, knowing the coordinates a peer is located at, we are able to identify the peer responsible for this location. With this knowledge at hand, area searches are possible. Assuming that one peer $p$ wants to find all peers located in a specific area (or other objects related to that area), peer $p$ has to do the following: first, peer $p$ has to calculate the highest z-key which is inside the search area. Second, peer $p$ has to lookup the DHT node responsible for this z-key and ask for all objects the specific node is responsible for. Knowing the DHT node's responsibility range, peer $p$ can calculate the value of the next highest z-key located in the search area. Doing so, peer $p$ can lookup all DHT nodes whose DHT identifier match with the calculated z-key inside the search area. After some visits, peer $p$ has gathered all information available inside the desired search area.

In our evaluation, we compare LobSter, our location-aware overlay consisting of two separated layers to Geodemlia which is not layered and rather a dedicated solution in the fields of location-aware overlays. We compare the performance of the different approaches in terms of missing nodes during area searches and number of visited nodes during area searches. We show that there is a tradeoff between lookup success and costs in terms of visited nodes that is contradictory in both approaches. Whereas lookups in Geodemlia might fail more often than in LobSter, less nodes are visited during an area search in Geodemlia. A problem which only occurs in Geodemlia is that lookups might not be deterministically because routing tables are not necessarily responding to changes in the network. It might happen therefore, that a peer $p$ has another peer $q$ in its routing table but not vice versa.

For our evaluation, we simulate a $20km$ x $20km$ wide area in which peers are randomly located. Besides different stationary scenarios in which peers are not moving, we simulate peer movement inside the $400km^2$ wide area. Peers follow a random movement model with a random velocity between $3.6km$ per hour (pedestrians) and $54km$ per hour (cars). The speed limit is chosen because it represents a typical speed in common cities which can be reached without getting fined. In our evaluation, it can be seen that lookups are more successful the smaller the search area is. Reason for multiple failures during area searches is the high fluctuation of peers in the respecting areas, caused by peer movement. In general we observe that existing location-based peer-to-peer overlays are not considering peer movement and would have to be extended enormously to support frequent changes of peers' locations Our solution, LobSter, on the contrary is aware of moving peers and achieves good results in area searches with a radius of approximately 4 percent of the simulated world. Area searches within this radius, produce the best result in terms of the ratio of found nodes to total nodes in the area, and within this radius, the costs of LobSter in terms of visited nodes during search are comparable to other specialized solutions, like Geodemlia.

## 6.2 Related Work

Solutions of location-aware overlays can be classified into mapping approaches which map locations to an underlying lookup service, e.g. a DHT, and dedicated overlays in which peers store information about physically close neighbors. Both categories of solutions have their advantages and disadvantages.

Solutions in the first category of location aware overlays benefit from a mapping function which allows to calculate the responsible peer for an overlay key instantly. Different techniques like space filling curves, used by Knoll et al. [85], the authors of CISS [87], or [31], are used as mapping function. Other approaches in this class use tree structures to map location to the overlay space. Examples are [90], [136], and [145]. DiST [100] uses indexing servers to map peers to areas. LIGHT [135], FAN [131], and further approaches [144, 79] use more specialized structures to maintain their routing tables. Problem of those solutions is often the fact that peers responsible for a specific location are far away in the underlying overlay. As a result, area searches are difficult to be conducted.

Although specialized location-based overlay solutions in the second category bring their own structure and guarantee node positioning near to locations, peers are responsible for, maintaining those overlays is cost intensive in terms of routing table updates. CAN [114], HyperCuP [123] are examples in which peers are linked to few neighbors only. Globase.KOM [86], RectNet [75] and GeoP2P [15] use tree structures to divide the geographical spaces into zones. GeoPeer [12] uses greedy routing through a triangle cell structure. Geo-Chord [109] combines Chord with a grid-based approach. Brambilla et al. [25] and Picone et al. [111] present an adaptive peer-to-peer overlay in which peers are stored in buckets, according to their distance to other peers. GeoKad [110] and Geodemlia [70] follow this approach.

## 6.3 Contribution

Our contributions in this paper are the followings. First, we introduce LobSter, a lightweight location-aware overlay which uses an underlying DHT as basis which can be exchanged at any time. We show how to utilize space-filling curves as mapping technique to provide area searches in LobSter. Second, we compare LobSter to Geodemlia and show that both overlays follow two distinct approaches. Dedicated overlays like Geodemlia have the problem that lookups might not be deterministically. Reason is that routing tables in these approaches might not respond to changes in the local network of a peer. Third, we evaluate our approach with moving peers. Current location-aware overlays are not considering peer movement and would have to be extended tremendously to support it.

## 6.4 Personal Contribution

The reviewing and selection process of the International Conference on Computing, Networking and Communications (ICNC) 2017 resulted in an acceptance rate of 29%. Tobias Amft, the author of this thesis, wrote the complete paper, designed LobSter, the location-based overlay solution, implemented and evaluated LobSter and a simplified Geodemlia protocol in Peerfact-Sim.KOM [1]. Kalman Graffi contributed with a review of the paper and discussions about the overlay solution and evaluation methodology.

## 6.5 Importance and Impact on Thesis

In this paper we show that a DHT as underlying basis can be utilized to solve more complex tasks, like distributed, location-dependent search. Introducing LobSter, we follow the idea that application and basic overlay can be separated into different modules. In this way we created a location-aware peer-to-peer overlay which consists of a basis DHT and an additional part which provides desired functionality for location awareness. Moreover, we consider a first approach that allows participating peers to move during the use of the overlay. Except for CISS [87], no other location-based P2P overlay based on DHTs considers peer movement. Compared to other location-based overlays, especially compared to Geodemlia [70], our paper [8] presents a location-based solution placed on top of existing DHTs which is aware of mobile peers changing their positions constantly.

Further we compare in this paper the advantages and disadvantages of two possible approaches of location-aware overlays. In general, we observe that in the category of location-based overlays two approaches exist. In location-aware overlays which base on a mapping approach, infrastructure is provided through an overlay which is not location-aware per se. Data objects in this approach (files or peer contacts) are usually associated with a physical location. Main characteristic of this category is that a mapping function exist, which maps the physical location of data objects to overlay identifiers of the hosting overlay. In this way, for every data object, a responsible peer in the hosting overlay can be found. Advantages of mapping approaches are on the one hand that responsible peer for a specific location can be calculated directly and that routing is provided by an underlying overlay, for example a DHT which provides routing in $O(\log(N))$ where $N$ is the size of the network. A major disadvantage of mapping approaches is that peers are not located in areas they are responsible for. Thus, peers responsible for neighboring space might be located far away from each other so that area searches are more exhaustive compared to algorithms that search the direct neighborhood of peers. Second effect of this issue is that peers responsible for certain location might be located far away from that location so that changes in the network are not noticed directly.

In the second category of location-aware overlays that we call the neighborhood approach, a dedicated overlay structure is established which is location-aware itself. Nodes in this overlay usually maintain links to neighboring nodes in their routing tables. Characteristic of this category is that nodes are responsible for the physical space around their own physical location. The main advantage of this approach is that peers are located around data objects they are responsible for. In this way, area searches and searches for the nearest neighbors only have to be conducted on peers which are actually located in the respective search areas. Nevertheless, one drawback of this approach is that responsible peers for a specific location can not be identified directly but have to be searched or looked up. Second and more important disadvantage of the neighborhood approach is that routing tables are limited in their size, so that not all peers in

[1] http://peerfact.com/

the direct neighborhood can be stored inside the routing table. As a result, routing tables of peers might be outdated or incomplete in crowded places.

With our paper [8], we present another P2P application which can be placed upon an existing DHT. Similar to Chapters 4 and 5, we show in this paper that different P2P applications can be built upon existing DHTs without the need to change them. This chapter also shows that new applications like DOSNs or location-based services, relying on an existing DHT, can be designed and implemented easily as extension of a DHT that is used commonly by all solutions. This observation allows us to reduce costs in the way that only one DHT is needed to support $N$ different applications. Having $N$ different applications running on one peer that share one DHT saves $N-1$ DHTs in comparison to operating $N$ applications that come with their own DHT. With Chapters 4 to 6, we show that a common DHT which is able to reduce the costs of the whole system can support multiple applications on one peer.

# Chapter 7

# A Tale of Many Networks: Splitting and Merging of Chord-like Overlays in Partitioned Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [7] which is a result of Tobias Amft's master thesis [6].

Tobias Amft and Kalman Graffi. "A Tale of Many Networks: Splitting and Merging of Chord-like Overlays in Partitioned Networks". *Technical Report: TR-2017-001.* Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany. 2017

Frequent join and leave procedures are considered in most works on overlay networks by default. On the opposite, few overlays only are aware of isolation events at national scale in the underlay. The disconnection of any country from the rest of the world leads to massive communication breakdown in the underlay. The Egyptian government for example ordered ISPs to delete routing entries of the BGP which connects Egypt to neighboring countries and thus, the Internet. Such an isolation event effects running overlays in unforeseen ways. As a result, overlay nodes located inside the isolated region could form their own separated overlay partition during the isolation. In contrast, other peers which are not inside the isolated region might be barely effected by the isolation event, depending on how sparse or narrow the isolated peers had been distributed over the network. Nevertheless, existing work on overlays barely considers the merging of separated overlay partitions during and after an isolation event. To the best of our knowledge, only two approaches considering the merging of separated Chord overlays exist, namely the Chord-Zip [82] and the Ring Unification Algorithm [125]. Ideally, isolated peers should re-structure the isolated overlay partition as long as they are disconnected from the main overlay. After isolation is finished, all separated overlay partitions should merge to a global network again. In order to reduce message costs, the number of started merger attempts is reduced to few nodes per connected overlay component only. As a result, the merging protocol is mainly active during isolation phases and reduces it costs during connected times to a minimum.

## 7.1 Paper Summary

We observe for Chord-like overlays, that during isolation multiple disrupted overlays parts are formed. Opposite to our expectations, isolation does not cause two distinct overlay partitions to be formed. Instead, we observe that multiple overlay *constructs* are formed during network partitioning events which are of the following three types. First, some nodes in the overlay might be strongly connected so that during isolation the ring structure can be recovered completely. This first type of constructs we call *ring*. Nodes in this construct form an own overlay which is complete in functionality. Second, a construct of loosely connected nodes could be formed that we call *chain*. In a chain, each node is connected to its successor node except for the first node in the chain whose successor pointer is empty or pointing to itself. Third, the node at the end of a chain might be connected to another chain or ring. This construct is called *hanger-on* as it is attached to other constructs. Although very slowly, Chord's stabilize algorithm normally integrates nodes in hanger-ons step by step into existing structures. For our classification we focus on successor pointers only, because only they have to be correct for successful routing. Fingers and predecessors are only additional pointers in Chord.

In this paper, we address the issues of separated peer-to-peer overlay parts stated above and present a new merging algorithm for Chord-like overlays which is supposed to unite different overlay constructs as soon as their participants are connected again in the corresponding underlay. In specific, the algorithm merges all constructs whose participants are still connected in the underlay during isolation, so that $N + 1$ Chord rings are formed, with $N$ being the number of regions isolated from the global network. As soon as isolation is over, the isolated overlay part is merged with the remaining constructs so that one global Chord ring is formed in the end. Our solution, the *Ring Reunion Algorithm* comprises different techniques which are necessary to achieve the desired goal. One part of our solution is a mechanism which extends the Chord routing table with additional, randomly selected overlay contacts. Those additional contacts can be probed for reachability in case of an isolation event. In order to reduce network traffic, the amount of started merger instances is regulated dependent on the size of an overlay construct. Doing so, nodes estimate the size of the construct they are currently in. Only a fraction of $\frac{\alpha}{size_c}$ nodes executes a merging attempt upon isolation, where $\alpha$ regulates the number of initiated merger instances per construct and $size_c$ denotes the size of the respective construct.

If a node in another construct is found, the Ring Reunion Algorithm starts to merge the two constructs consequently. To do so, a newly found peer $p$ is compared to the own successor $s$. If peer $p$ is better suited as successor, $s$ will be replaced by $p$. Node $s$ is thereupon sent to $p$ which is comparing $s$ to its own successor. In case a peer $p$ is not better suited as successor, it is passed to the next successor until it is better suited. In this way, the Ring Reunion Algorithm compares an alternative successor locally to the existing successor, always selecting the best suited peer as new successor. As a result, two distinct constructs are merged to one structure again. The Ring Reunion Algorithm automatically stops if a construct is discovered to be merged already. To speed up the merging process, the Ring Reunion Algorithm is able to run multiple instances in parallel. Doing so, the Ring Reunion Algorithm instructs the furthest $k$ fingers of a Chord node to start a merging instance simultaneously. As a result, the time to reunite the separated constructs is halved each time the number of parallel instances is doubled.

To evaluate the Ring Reunion Algorithm, we compare our solution to existing merging algorithms, namely Chord-Zip [82] and the Ring Unification Algorithm [125]. First, we investigate the algorithm's basic functionality to merge multiple rings. Second, we consider different techniques to identify possible merging candidates. Additionally, we compare our algorithm with related work in a complex realistic scenario in which multiple regions become isolated. We show that too many parallel merging instances can cause a negative result as well since to many constructs are ripped apart by the merging algorithm. Therefore, we identify good parameter settings for our algorithm in different scenarios. Our evaluation shows that out of the presented overlay merging algorithms, only the Ring Unification Algorithm [125] and our Ring Reunion Algorithm are able to merge multiple separated rings reliably. The Chord-Zip [82] algorithm fails to merge multiple rings in reasonable time. In complex scenarios, in which multiple regions are isolated, only our Ring Reunion Algorithm is able to merge the partitioned overlays again. To reduce message costs, our Ring Reunion algorithm reduces its activity during stable periods and starts to merge separated nodes as soon as isolation events occur.

## 7.2 Related Work

Data probagation, stabilization, restructuring and merging processes play a great role in distributed networks like peer-to-peer overlays, MANETs, and opportunistic networks. Different works give an overview on issues in opportunistic networks [29], or data-propagation in wireless mesh networks [98, 63, 97]. For peer-to-peer overlays, different approaches are needed.

In the first category of overlays, solutions are introduced which consider changes in the underlay to reassemble overlay structures. Jain et al. [76] introduce a routing algorithm which leverages information about the underlying topology. Musolesi et al. [99] connect MANETs through bridging nodes between distinct overlay partitions. MaxProp [28] gathers knowledge about the underlying network for routing in vehicle-based delay-tolerant networks.

The second class of overlays comprises techniques to stabilize peer-to-peer overlays in the presence of network partitioning events. Datta et al. present in [42] typical issues for the merging of ring-based overlays. In [41], Datta investigates Chord's performance during the process of merging two Chord rings by summarizing the robustness of routing in Chord during a merger. Chord-Zip [82], [83], and the Ring Unification Algorithm [125] are two practical solutions for the merging of Chord-like overlays.

An overview on how P2P overlays and DHT-based routing are applied to MANETs is given by Abid et. al in [3] and [2]. Shah et al. discuss the merging of structured and unstructured P2P networks over MANETs in [126] and [127]. In [4], Abid et al. present methods to detect different logical networks and to merge DHT-based logical networks in MANETs. Heer et al. [73] describe the combination of DHTs and MANETs and focus on problems with network partitioning events in the underlying MANET.

The third category of solutions comprises approaches which stabilize autonomously from any network state. Re-Chord [84], Ca-Re-Chord [21] and HSkip+ [57], are self-stabilizing overlays, which create desired overlay topologies from any random, connected graph. Earlier in 2005, Shaker et al. [128] proposed a protocol to let peers, which are connected in an arbitrary state, converge to a structured peer-to-peer overlay based on a ring structure.

## 7.3 Contribution

With this work, we discuss different design approaches of existing merging algorithms for Chord-like overlays. With our Ring Reunion Algorithm, we extend existing state of the art approaches with a new solution. Further, strengths and weaknesses of existing overlay merging algorithms are investigated and compared against each other.

In general, our proposed merging algorithm can be used as additional process to stabilize Chord-like overlays. Isolation events are similar to heavy churn during which lots of participants leave the network. Participants in Chord maintain usually a list of successors to survive massive leave events. Merging algorithms extend Chord's robustness with the ability to repair broken overlay parts quickly.

## 7.4 Personal Contribution

Tobias Amft wrote the paper, designed the Ring Reunion Algorithm, and evaluated it against the existing Chord-Zip and Ring Unification Algorithm in PeerfactSim.KOM [1]. Kalman Graffi contributed with various ideas in the creation of the overlay. He further contributed with the guidance on the evaluation methodology and the review of the paper.

## 7.5 Importance and Impact on Thesis

This work answers the research question on how to merge split communication islands after network partitioning events. We show with the Ring Reunion Algorithm a possible way to realize merging algorithms for Chord-like overlays. Overlay stacks using a Chord-like overlay as basis DHT benefit from the merging algorithm in the way that robustness of the underlying DHT basis is massively increased, especially in the presence of isolation events and other massive, unexpected churn events.

Merging algorithms have a huge impact on overlay stacks as they are described in Chapter 1. First, all P2P applications built atop a common DHT, in this case Chord, benefit from a merging algorithm which updates the common DHT's routing table permanently. As all applications on top of the common DHT share the same routing table, they are affected by the results of a merging algorithm likewise. Second, a single merger algorithm designed for a specific DHT can influence other DHT-based and non-DHT-based overlays as well if they are connected through a common routing table in the overlay stack as described in Chapter 1 and Chapter 8. The purpose of the common routing table is that overlays can share their contacts

[1] http://peerfact.com/

with other overlays. If a merger algorithm updates contacts from one specific overlay, other overlays in the overlay stack benefit from the updated routing table as well. In Chapter 8 we further discuss the idea of a common routing table and evaluate the coexistence of multiple overlays sharing one routing table.

# Chapter 8

# The Benefit of Stacking Multiple Peer-to-Peer Overlays

This chapter summarizes the contributions and gives a verbatim copy of our paper [9].

Tobias Amft and Kalman Graffi. "The Benefit of Stacking Multiple Peer-to-Peer Overlays". *Technical Report: TR-2017-002.* Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany. 2017

In the last decades, many different peer-to-peer overlay applications have been proposed as alternative to traditional client-server applications. Distributed applications serve a small niche in today's plethora of communication tools, especially to preserve anonymity and privacy, as stated in Chapter 2. Under normal conditions, coexisting overlays are not aware of each other, instead, each overlay has been designed for one special and individual task which usually does not enable cooperation with other overlays. Nevertheless, it could happen that multiple overlays coexist on a single peer, especially if challenging overlay solutions are require a combination of coexisting overlays to solve a certain problem. Overlay synergies have been barely considered in recent work, although a symbiosis of coexisting overlays can help to stabilize the overall system of overlays running on a peer.

In Chapters 4, 5, 6, and 7 we propose different P2P applications as extension on top of an existing DHT. The main idea behind the basic DHT is that multiple applications could access a shared DHT which provides a core lookup service to all applications. In Arsham Sabbaghi Asl's master's thesis [16], we operated the four applications on top of a common DHT to analyze the behavior of DHT-based applications. It can be seen that a common DHT reduces overhead related to duplicate lookup functionality, but also influences applications as they are coupled through the common DHT. Nevertheless, although a common DHT provides basic lookup functionality, it will never be able to serve all requirements of all applications on top. Especially if applications have certain demands to an underlying lookup or search overlay, an additional core set of functionalities, which is not provided by the DHT, is needed.

In our paper [9], we investigate further possibilities to combine overlays and relative overlay parts with respect to reduce implementation and operating costs. Further, we describe in detail the basic modules most overlays are made of and we identify which redundancies can be avoided if overlays coexist on a single peer.

## 8.1 Paper Summary

In this paper, we deepen the knowledge about coexisting overlays through a classification of several possible overlay synergies. Furthermore, we focus on the description of core functions of peer-to-peer overlays and identify an overlay specific and an application specific part of peer-to-peer systems. Most important, we describe the idea of an overlay stack and we show that coexistence of multiple overlays leads to increased robustness of the complete overlay system.

Maniymaran et al. observe in their work [92] a primary and a secondary component in both structured and unstructured overlays which relates to overlay contacts in the routing table only. We further identify general components of common overlays which allow individual optimizations. In conclusion, overlays consist of three important types of modules which in specific constellation form the desired behavior of an overlay. Those modules are the overlay *routing table*, the overlay-specific *routing algorithm*, and specialized *update mechanisms* to keep the overlay topology stable and up to date. Each overlay maintains overlay contacts, which hold information about other peers in the network, in a routing table. Peers are usually sorted according to an overlay-specific identifier inside the routing table. The routing algorithm in turn implements an overlay-specific routing mechanism which forwards messages between any pair of peers. In comparison to the routing table, the routing algorithm is an active module which requests known contacts from the routing table. Due to churn, overlays might become instable after some time. Diverse update mechanism are required to stabilize an overlay in the presence of churn.

In this paper, we identify three possible categories of coexistence between overlays. In the first category, different overlays exist on one peer without knowing each other. No changes of overlay structure or implementation are required in this category, because desired functionality is fully provided by each overlay. Depending on existing implementations, new functionality has to be implemented in new overlays rather than introduced as extension of existing protocols. Each overlay is independent from other implementations so that new overlays can be added easily without effecting the remaining overlays. The performance of each overlay is correlated to the resources (bandwidth consumption, storage consumption, CPU usage, etc.) offered by the hosting peer.

Category two of overlay synergies comprises all overlays which are build on a common base overlay, which is mainly a DHT that provides a basic lookup service to other applications. An additional unstructured overlay can support the common base overlay with search functionality. In this class, a strict separation of peer-to-peer systems into the components overlay and application is required. Whereas the overlay part provides communication, on or more applications on top introduce specialized functionalities. Redundant functionalities are avoided and basic costs are reduced to the costs of one DHT. Further functionality can be added as new application on top of the common base overlay, so that implementation costs for new features are reduced. In Chapters 4, 5, 6, and 7 we prove that many contrasting applications can be realized on top of existing DHTs without a general loss of functionality. Nevertheless, the problem with one basic DHT only is, that all applications need to fit the requirements of the utilized base DHT. Applications which demand more specialized routing mechanisms can not be served fully by a common DHT.

To overcome the limitation of a base common overlay, the third category of coexisting overlays

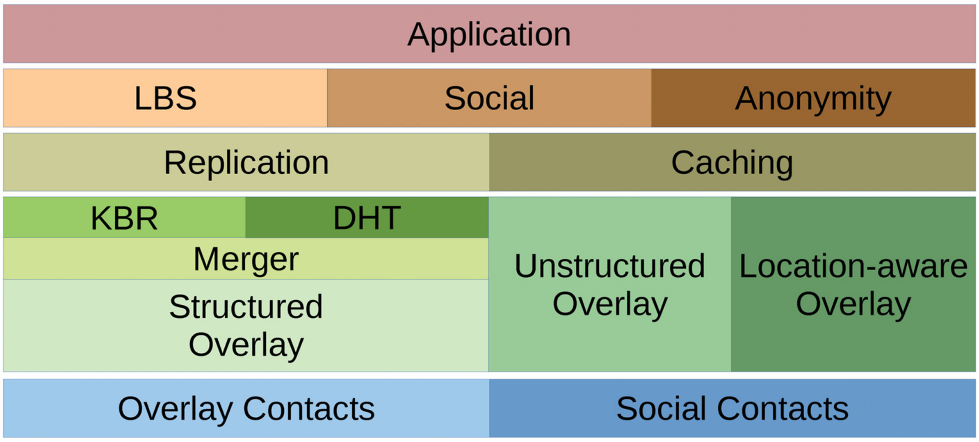


Figure 8.1: The overlay stack can be enhanced in any direction.

separates overlays into small modules, each fulfilling one special task. The combination of different modules leads to a desired overlay in which redundant functionalities can be avoided immediately. In fact, a common base DHT as lookup service can be also realized through the stacking of different overlay modules. Additional overlay functionality can be added as new module, mostly without the need to re-implement existing modules. Consequently, each module requires clear interfaces and a strict modular implementation of new features so that combinations of different implementations are possible.

In our paper [9], we introduce the concept of an overlay stack which is the combination of different specialized overlay modules to obtain one or multiple overlays or overlay applications with the goal to avoid duplicate implementation or operating costs. We introduce the concept of a shared routing table for our overlay stack, which allows all overlays executed on one peer to add contacts into a common routing table and to obtain contacts from it. Additionally, we add a base communication layer which decouples the overlay modules from the actual networking functionalities provided on a host. As a result, peers can send their own identifiers piggybacked on overlay messages to neighbors. Doing that, peers learn about all identifiers of their surrounding neighbors. Figure 8.1 drafts, how different overlay modules could be stacked together to an overlay stack. It can be seen, that the overlay stack consists of different specialized modules like those stated in category three. Nevertheless, parts of the overlay stack could consist of a basic DHT serving different applications.

In our evaluation, we execute Chord, Geodemlia, and a Gnutella-like overlay in parallel on each simulated host. In all three overlays random lookups or searches are started frequently to simulate user behavior. Moreover, peers permanently join and leave the network, leading to an instable state. Comparing unmodified versions of Chord, Geodemlia, and Gnutella to an overlay stack, in which the three overlays share a common routing table and communication layer, we observe in our simulations that

## 8.2 Related Work

Only few works consider the coexistence of multiple overlays on one physical peer. Maniymaran et al. [92] identify in their work two types of components in peer-to-peer overlays, the primary, and the secondary component. The classification describes the importance of different parts of the routing table to a respective overlay. The primary component of a routing table is inevitable for the correct operation of an overlay. However, the secondary component of an overlay which provides only optimizations of the base overlay can be replaced with existing components which are known from other overlays. In this way, coexisting overlays benefit from the components of other overlays so that only parts of each overlay have to be implemented to obtain full functionalities of all coexisting overlays. Lin et al. [89] categorize common synergies of gossip-based overlays and demonstrate their benefits. Other related work can be found which is distantly related to the topic of coexisting overlays.

The key based routing (KBR) API [40] for example is an early description of fundamental functionalities for structured (and partially unstructured) overlays. It lists which basic functionalities an overlay should provide. Replication services like PAST [54], or publish /subscribe approaches like SCRIBE [119] can be built on top of overlays which implement the common key based routing interface.

A common approach to separate applications from the underlying routing logic is to build applications on top of existing DHTs to make them independent and interchangeably. As an example, Chawathe et al. present in [31] a case study on how to build applications atop existing DHTs. In [62, 67, 64, 65] multi-layer frameworks for social networks are presented and discussed. Graffi et al. for example present in [67] a distributed data plane on top of the DHT Pastry [118]. Adaptive Chord [142] is an example for an overlay structure to support capacity-aware search on top of underlying DHTs which allows content-based search. Monitoring approaches such like [68, 66, 113] utilize underlying DHTs to capture desired metrics in a running overlay. The gathered information could be used then to adjust different parameters in a DHT with the aim to optimize the parameter setting and thus the resulting quality of the network.

Other solutions, such as OCALA [78] and Oasis [91], introduce new layers in the network stack which enable to connect different overlays and applications. A special layer is introduced in these works in order to multiplex and demultiplex traffic which is sent between different overlays. Oasis [91] even allows to use overlay-based packet delivery services in parallel to regular IP-based routing.

SLOSL [19] and MACE[1] [81] (extension of Macedon [117]) are approaches to define an overlay describing language. With those approaches the creation of overlays should be simplified. Whereas MACE produces executable finite state machines which make it hard to extend and reuse existing overlay implementations, SLOSL misses an evaluation of the quality of the resulting overlays. Furthermore, both works only consider creation of single overlays in their discussions. Both works do not mention how to combine overlays or how to reuse existing modules.

[1] http://www.macesystems.org/mace/

In the last category of solutions, approaches combine different applications into one overlay. They usually combine multiple features which are difficult to operate separately. In [33], anonymity in location-based overlays is preserved by using spatial cloaking mechanisms to blur the exact location of a user in a location-based, mobile peer-to-peer overlay. In [1], overlays change their structures to match the underlying network.

## 8.3 Contribution

In this paper, we identify typical parts of common overlays and characterize different synergies of coexisting peer-to-peer networks. We discuss interdependencies and possible optimizations of the proposed overlay modules. Specifically, we identify a core set of modules required from any overlay, we detect redundant functionalities in coexisting overlays, and we show that coexistence of overlays can result in more robust overlays.

## 8.4 Personal Contribution

Tobias Amft, author of this thesis, wrote the paper, identified and described a core set of modules in existing peer-to-peer overlays, and implemented modular versions of Chord and Geodemlia in PeerfactSim.KOM [2]. Kalman Graffi contributed with frequent discussions about stacked overlays and he reviewed the paper. Special thanks go to Newton Masinde, who reviewed this paper at the very last moment.

## 8.5 Importance and Impact on Thesis

In the previous Chapters 4, 5, 6, and 7, we proved that many contrasting P2P applications can be built on top of existing DHTs without loss in functionality, in comparison to specialized applications of the same type. We identify a common DHT to be a slim core set of functionalities which can be used as good basis for diverse P2P applications. We showed theoretically and practically in our papers [72] [11] [8] [7] and in Arsham Sabbaghi Asl's master's thesis [16], that it is possible to implement different applications on top of a common DHT so that duplicated functionalities are prevented and redundant implementation and operating costs are avoided. Doing this, we notice that one common DHT defines a good core set of functionalities for diverse P2P applications, but is not sufficient to serve all demands of all possible applications, especially in cases in which more specialized routing mechanisms are required. Having one DHT as basic lookup service, we reach our limits to optimize the coexistence of different P2P applications very fast.

In this chapter and in our paper [9], we therefore investigate the characteristics of basic routing overlays to find out what those basic overlays like Chord, Pastry, Geodemlia, etc. are made of

[2] http://peerfact.com/

so that we can derive a methodology to build any desired P2P overlay or application. We find out, that routing overlays mainly consist of three modules, a routing table, in which overlay contacts are stored, a routing algorithm, which defines rules to forward messages, and different update mechanism, which update the overlay's routing table periodically. Characterizing the different parts of an overlay, we identify a basic core set of functionalities for different routing overlays, like DHTs, location-aware overlays or unstructured overlays. Summarizing the overlay modules, it can be seen that only the routing table of an overlay is a passive module which gets accessed by other modules. Therefore, we propose the idea of a common routing table as basis for for different routing overlays. Specialized update or routing modules can be added to the common routing table in order to form a certain behavior.

The combination of the common DHT approach with a common routing table and the modular implementation of overlay modules results in an overlay stack which is to combine different modules to obtain one or multiple overlays or overlay applications. In other words, the overlay stack is the idea to combine different overlay parts in a way that a certain behavior is obtained and second, that the implementation of duplicate modules and functionalities is avoided. Inside the overlay stack, a common DHT or any other routing overlay might propose a basic lookup service for diverse applications, whereas a common routing table constitutes a core module for different overlay routing algorithms and update mechanisms. In summary, Chapters 4, 5, 6, and 7 show how our overlay stack can grow vertically by applying new applications on top of existing overlays or applications, whereas this chapter extends the overlay stack horizontally through the enhancement of new core routing functionalities.

We show in our paper [9] that multiple coexisting overlays sharing one common routing table can be executed on a single peer without loss of quality and performance, compared to the original overlays which are operated separately without common routing table or other shared modules. In addition, we show in our evaluation that the use of a shared routing table increases the robustness of all participating overlays while operating costs and implementation costs of the overlays are reduced. We further show that an overlay stack allows to switch on and off new routing functionalities during runtime, since the common routing table supports bootstrapping new overlays without the need of full join procedures.

# Chapter 9

# Conclusion and Future Work

Different peer-to-peer overlays and distributed applications have been investigated in the past decades. Typically, each new overlay presented is a specialized solution for a given problem that is to be solved in a distributed manner. The combinations and parallel usages of different overlay implementations have not been examined and understood yet, since they are rarely in focus of research. Although P2P systems became known by distributed file-sharing solutions for fast downloads and fast information dissemination without the need of an underlying infrastructure, we believe that nowadays, P2P systems can fill different niches in social communication tools. Reason is, that centralized communication tools can be blocked, censored, and monitored easily by attackers like oppressive governments, whereas P2P systems are harder to control completely, due to their distributed design and infrastructure which lacks a central point that could be shut down to turn off the whole system.

The motivation behind this thesis are various research questions which extend the view on existing overlay solutions by finding similarities and common patterns for overlay networks. For deeper investigation, we focus on diverse overlay applications which require, due to their diversity and contrasting intentions, specialized and distinct solutions. In specific we compare overlays which either belong to social centric approaches or to location centric approaches. We consider social centric approaches as social networks are an interesting and modern topic and divide those approaches further into restricted and open communication. Also the location centric approaches which are especially relevant for mobile devices are subdivided in disturbed and free communication.

We are mainly interested in the implementation of different solutions, that we limit to four contrasting use cases, to evaluate and analyze common patterns the overlays use. Goal of this thesis is to identify basic requirements the different applications demands. With this knowledge at hand, we answer the important question what overlays are made of. We then investigate the parallel execution of multiple overlays on a single peer. It should be possible to reduce costs by defining a common basis that is placed underneath specialized overlay solutions. Functionality which is commonly used should be provided by one basic overlay only. Specialized solutions should be placed upon the basis without the duplication of existing functions. We propose basic overlay functions and show that they can be realized as small modules which can be assembled at will to form a certain overlay.

In the following, we conclude our findings about the proposed applications and summarize the derivation of an overlay stack which provides basic functionality for peer-to-peer overlays and which is extendable at any time.

## 9.1 Conclusion

Most services running over the Internet rely on a background infrastructure made of one or multiple servers. Clients which consume the specific service connect to a server and send data and user information to the remote device. We believe that the amount of services running over the Internet will further increase over the next years. Due to insights in big data analysis, grid and cloud computing, and further modern techniques, resources like bandwidth and CPU are not a limiting factor for server-based solutions. Still a limiting factor which speaks against client-server approaches is that user data is stored at a single service provider which is often located abroad so that the users lose control over their data. Another point is that server-based applications are easier to block or monitor than distributed equivalents. We think peer-to-peer applications are still an interesting topic in the next years since data in peer-to-peer networks can be maintained costly and under control of its users. In this thesis, we focus therefore on the efficient building of peer-to-peer overlays for different and diverse use cases.

- In Chapter 1, we introduce and motivate the idea of an overlay stack, which is to implement basic overlay functionalities in small modules which can be stacked together to form a full overlay. In this way, duplicated implementations and usages of functionalities are avoided in comparison to the naive approach of having multiple overlays executed in parallel. We motivate four contrasting application cases which help us to investigate possible designs of an overlay stack empirically. We propose open research questions related to stacked overlays and formulate goals for this thesis, out of which one is to classify basic overlay functions and to find out possible redundancies in the assortment of overlay functions. The chapter is closed with a list of contributions and an overview about the overlay stack. In this thesis, we evaluate the overlay stack approach to show that it reduces implementation costs, costs at runtime, and is more is more efficient than the parallel execution of plain overlays.

- In Chapter 2, we review political events which took place around 2010 and 2011 in the Middle East and Northern African (*MENA*) region. In specific, we review technology and infrastructure used in this region for communication during the *Arab Spring* uprisings and we discuss the technical reactions on side of African governments during the revolts. The shutdown of Egypt's connection to the global Internet is one extreme example that regimes and governments are interested in the controlling of Internet traffic, e.g. to enforce censorship and control over Internet users. We review modern techniques which are used by governments to censor network traffic and discuss possible countermeasures which foster private and anonymous communication over the Internet. We extend the article with a discussion about possible influence the Arab Spring had or still has on today's academical thinking. We believe that the political situation during the Arab revolts did not lead to a significant change in academic thinking in the field of computer science, but it constitutes a great example, that communication in some countries of the world is endangered through arbitrary and unforeseen actions of certain regimes. Distributed communication tools which are not dependent on a remote server somewhere in another country, which provide their own communication infrastructure are most likely able to survive shutdown events on national scale.

- In Chapter 3, we summarize the three network types: peer-to-peer overlays (*P2P*), mobile ad-hoc networks (*MANET*) and opportunistic networks (*OppNet*). We discuss possible scenarios in which the combination of all three network types could be useful. We come to the conclusion that a combination of the three proposed networks is considerable in situations in which OppNets and MANETs would provide a basic infrastructure for communication, so that P2P overlays could be used for specialized infrastructure and data organization atop. Those scenarios might include emergency cases or distant places without proper communication infrastructure. Important for this thesis is the identification of a suitable peer-to-peer simulation tool in which our protocols and ideas can be tested. We tested and summarized nine different simulation tools out off which two have been identified as possible candidates for our simulations. OverSim[1] is a flexible simulation framework built upon the general simulation framework OMNeT++[2]. It benefits from a great variety of overlays already implemented. PeerfactSim.KOM[3] on the other hand has a more layered structure which allows to extend the simulator with specialized modules easily.

Our goal in this thesis is to understand the construction methodology of overlay networks in order to identify and point out core functionality and elements required for an overlay. Through this deep understanding we propose a construction scheme that allows to combine various desired functionality in a combined overlay network. The combination of current overlay implementations is desired to obtain different functionalities from various overlay solutions at once. Unfortunately, the combination of overlays is currently limited to the parallel execution of existing implementations. A more enhanced approach in which overlays and applications are implemented as small modules on top of existing modules reduces implementation costs and costs at runtime enormously. If for example multiple distributed applications share a common lookup service, like a DHT, resources and bandwidth needed to maintain the lookup service can be saved, since only one service instead of x different instances of the same service have to be executed. Second, the combination of small overlay modules which can be accessed by other modules gives a gain in functionality in the way that overlays benefit from additional services which are originally built for other modules. Overlays sharing one routing table for example benefit during their join phase from contacts that have been collected from other overlays already.

We focus in this thesis on four example applications which are diverse and contrasting in both, their purposes and their requirements. Distributed online social networks fulfill social centric requirements and provide usually unrestricted communication to its users. Anonymous peer-to-peer overlays on the other hand restrict communication between different users to hide the real identities of its participants. Location centric overlays are interesting research objects since mobile devices, such as smartphones, became highly popular in recent years. Location-awareness and general communication is endangered during partitioning events in the underlying network. Self-stabilizing algorithms are therefore valuable techniques to stabilize existing overlay infrastructures. In the chapters listed below, we propose and evaluate solutions for the four cases. Each solution is either based on an underlying DHT that can be replaced by any other DHT, or extends existing DHTs to become more robust during partitioning events.

[1] http://www.oversim.org/
[2] https://omnetpp.org/
[3] http://peerfact.com/

By building each application atop an existing DHT, the advantages of existing solutions like routing in $O(\log(N))$ (with $N$ denoting the size of a network) are reused for more specific purposes. Within the chapters summarized below we also characterize typical core elements of existing overlays and learn about the methodology how overlays are constructed so that we are able to optimize our overlay stack.

- In Chapter 4, we propose and evaluate a distributed online social network which uses a DHT as underlying lookup service to ensure availability of user data. In this work, peers delegate the task of providing their user-related data to friendly nodes while they are offline. Peers which are selected as replica nodes and hold user-related content are called points of storage and are indexed in the basic DHT lookup service so that they can be publicly found by interested participants. In this way, a data-availability service is generated which assures user-data to be available online, even if its owner is not available itself. While the basic DHT serves as a lookup service to find peers storing desired content, the application itself manages updates of content and user authentication. For the point of storage selection, we follow the idea of Robin Dunbar, a social scientist who discovered the number of actively maintainable relationships to approximately 150. This Dunbar number limits the number of known and trusted peers in our application to 150 peers likewise. In this way, peers consider a small subset of participants as trusted nodes that are potential candidates to store personal data. A tie-strength, related to the personal interaction frequency between peers, is used as measure for the trustworthy and reliability of a friendly peer and determines the election of a point of storage. In this paper we show that distributed online social networks can be realized as specialized application on top of existing overlays without the need to change the underlying lookup service to fulfill our purposes. We further show, that with our approach, it is possible to enhance DHTs with social-relationship-based replications techniques which enable more than 95 percent of all data to be online permanently with 2-3 replicas only.

- In Chapter 5, we investigate solutions for trusted routing algorithms over friendly nodes. Since routing via friends allows a certain degree of anonymity, our goal is to route messages through the network via friends only. The specialty in our approach is that we assume friendship relationships to be formed according to Dunbar's idea of ego-networks in which nodes have up to 150 friends only. Friendly contacts are further divided into four closeness groups, describing the personal trust a peer has in another contact. The challenge in the resulting ego-networks is that friendship can not exist mutually for every contact in every scenario. For example, if one node $p$ has 150 contacts and another peer $q$ is about to join his ego-network, node $p$ has either to decline joining peer $q$ or has to drop one of the existing friends. In both cases, the rejected node may hold a link to peer $p$, but no link is maintained vice versa. We follow in Chapter 5 the idea to extend existing DHTs like Chord [133] and Pastry [118] with the possibility to route via selected friends only. Sadly we learn with this approach that routing via friends is not possible with existing routing algorithms in structured overlays. The main problem is that in structured overlays certain structures are necessary to make routing effective. Considering friends only for routing does not fulfill the requirements structured overlays claim to make routing possible. With *goLLuM*, a novel routing algorithm which can be placed on top of DHTs, we show hat social-relationship graphs can be integrated into existing DHTs, so that routing via trusted nodes is granted.

- In Chapter 6, we focus on the implementation and evaluation of a location-based peer-to-peer overlay built upon an existing DHT. Our specific goal is to build a location- and motion-aware peer-to-peer overlay which uses a mapping function to map two-dimensional physical locations to a one-dimensional identifier space. Mapping coordinates to an overlay identifier allows us to store location-centric data directly in an underlying DHT like Chord [133] or Pastry [118]. On the other hand, the mapping function allows us to lookup data related to specific coordinates directly. We compare our approach *LobSter* to the existing Geodemlia [70] protocol and show that there are in general two distinct approaches to solve the requirements of location-aware peer-to-peer overlays. On the one hand, such overlays can be realized with mapping approaches in which coordinates are mapped to positions in an underlying, structured overlay. In this way, lookups to desired data can be started directly thanks to the underlying DHT. These approaches typically have the problem that peers are not located near the area they are responsible for. As a result, multiple lookups have to be started to perform an area search. In specialized approaches on the other hand nodes are usually responsible for locations around their own position. However, the main drawback comparing to mapping approaches is that data around a location has to be searched actively, direct lookups are not possible. Our evaluation shows, that our solution *LobSter* enables achieves good results in area searches withing a radius of approximately 4 percent of the simulated world, without being costlier than comparable solutions. Moreover, with the paper presented in Chapter 6, we find out that it is not possible to implement every application on top of existing DHTs. We notice that we have to consider specialized overlay functionality for the building of an overlay stack. An overlay stack should be extendable at any time.

- In Chapter 7, we compare different algorithms that are supposed to merge disrupted Chord-like overlays. We investigate the effect of partitioning events of national scale on Chord overlays. We show that during an isolation event in the underlay, multiple communication islands are formed in a peer-to-peer overlay like Chord. Peers which are not directly connected in an overlay will be separated during isolation. Our goal is therefore to investigate approaches that merge disrupted structures during and after isolation, so that after some time, a global or initial overlay state is reconstructed. The investigation of existing approaches like Chord-Zip [82] and the Ring Unification Algorithm [125] leads us to a novel solution, the Ring Reunion Algorithm. We focus in this approach on different necessary properties a merging algorithm should have to operate fast and reliably. We show in our paper [7], that our Ring Reunion Algorithm is able to merge separated overlays during and after isolation events in the underlying network while message costs are kept low in inactive and in active phases. With the work presented in Chapter 7, we learn that each peer-to-peer overlay needs an own specialized merging approach, because every overlay has its own functionalities and requirements to a merging algorithm. Due to the diversity in existing peer-to-peer overlays, there can not exist one merging algorithm fitting all existing overlay solutions. Limiting the overlay stack and a resulting overlay core to Chord-like overlays only would reduce our idea of a flexible and extendable basis enormously. Therefore, we focus in the following on the extension of the overlay stack with arbitrary overlay protocols.

In Chapters 4 to 7 we focused on four different solutions for contrasting overlay applications. Since our goal in this thesis is to find a core set of overlay functions and a resulting basis overlay which supports an easy stacking of multiple overlays on top, we concentrate in the previous chapters on applications built on top of existing DHTs. We find out, that multiple overlays can be implemented on top of existing DHTs but that those solutions are also limited to the characteristics of underlying DHTs. For this reason, we extend our insights about the stacking of overlays with a new concept to build coexisting overlays efficiently. Deeper investigations of various peer-to-peer overlays allow us to describe an extendable core set of functions that can be used to build any overlay irrelevant whether the overlay relies on an existing DHT or needs special structures. Although each overlay has its special characteristics, we observe similar patterns in every overlay that can be reused. Our goal is to describe the similarities of different overlays so that different approaches can be implemented through the stacking of small, task-tailored modules. Depending on the purposes of the different module, they can be exchanged with other modules.

- In Chapter 8, we discuss similarities in current overlays and describe our idea of a flexible and extendable overlay stack. First, we identify typical modules necessary in routing overlays. We identified them to be routing table $RT$, routing algorithm $RA$, and diverse update mechanisms $UP$. The routing table is a data structure in which known contacts to other participants are stored. Usually, the contacts are ordered considering a certain scheme, e.g. according to their distance to a certain node or simply in ascending order (considering overlay identifier). Fact is, that it is possible at any time to order a set of overlay contacts according to a certain routing table scheme. A routing algorithm determines how messages are routed in an overlay. Therefore, each routing algorithm decides to which contact or contacts a given message should be forwarded. To select a possible next hop, the routing algorithm needs to access the contacts stored in the routing table. Since the routing algorithm is main part of the overlay in which communication appears, it can be used to update knowledge about other peers, e.g. by overhearing communication with other nodes. Different additional update mechanisms can be used to stabilize an overlay. Those update mechanisms typically use the routing algorithm to update and disseminate knowledge about the network. In many cases it appears that update mechanisms act autonomously by contacting known contacts directly. Modules for joining the network also belong to the category of update mechanisms since their purpose is to fill a routing table initially on start of an overlay instance. Following the patterns proposed in Chapter 8, we are able to combine different overlay modules to new overlays and protocols. Existing functionalities can be combined or simply extended to new modules, since each functionality is implemented in a single module. In this way, duplicated implementations are avoided and general costs for the implementation of new functionality is reduced. We show that a positive side-effect of shared routing tables is that contacts are updated through multiple simultaneously running update mechanisms. In this way, overlay modules which are seldomly used or modules which have lazy updating mechanisms benefit from the maintenance strategies of other modules. Additionally, single update mechanisms can be adjusted so that less traffic is consumed to keep overlay structures updated. Another positive side-effect which can be seen in our evaluation is that the common routing table offers already known contacts for bootstrapping to other overlay modules in the overlay stacks, so that networks can be joined without effort and additional routing overhead. All in all, through the parallel execution of multiple overlays, each single overlay needs less resources in comparison to a standalone version of this specific overlay.

In this thesis, we identify core functionalities of P2P overlays and applications and describe a new methodology to create coexisting overlays while redundant implementation and operating costs are avoided. We call this new approach of coexisting overlays *overlay stack*, because overlays in this solution consist of specialized overlay modules which are combined to obtain a desired overlay or application. The advantage of the overlay stack in contrast to the parallel execution of existing overlays is that duplicate modules can be avoided and existing modules can be reused, exchanged, or optimized individually. To support our approach, we propose novel P2P overlay applications from contrasting fields and build them on top of an existing DHT. We show that a shared overlay, like a DHT, defines a good basis for a rich variety of P2P applications which can be used to organize coexisting overlays on a single peer. As we notice that one DHT alone might not fully define a core set of overlay functionalities, we propose further ways to extend our overlay stack. We find out in Chapter 8, that overlays consist of mainly three core modules, a routing table, a routing algorithm, and additional update mechanisms. The introduction of a common routing table allows different overlay modules in the overlay stack to access a shared pool of overlay contacts. Comparing our overlay stack to existing overlays running in parallel, we show that we are able to reduce operating costs and to increase the robustness of participating overlays in our overlay stack. We further show that an overlay stack allows to add new overlay functionality on top of a common routing table or on top of existing modules and allows to switch on and off new routing functionalities during runtime so that the bootstrapping of new overlay functionality is possible without costly join procedures.

In conclusion, we aim in this thesis to optimize the coexistence of multiple overlays running in parallel on a single peer. Pursuing our goal, we are able to answer our research question proposed in Chapter 1.2. We show that the quality of coexisting overlays and applications can be improved through the use of our overlay stack approach while different costs are reduced. At the same time, we identify requirements and common patterns of current P2P overlays and derive a methodology to build future overlays efficiently and sustainably. Our works presented in Chapters 2, 3, 4, 5, 6 ,7, and 8 lead us step by step to the development of the overlay stack, while each chapter by itself lets us understand the nature of coexisting P2P overlays a bit more.

## 9.2 Future Work

During the work on peer-to-peer overlays, we came in touch with many interesting research topics. As we are not able to cover all of them with one thesis, we present them here as possible future work.

### 9.2.1 Quality of Service and Adaptive Peer-to-Peer Protocols

Nowadays, a huge diversity of devices is used to participate in the digital world. The new technology shifted workspaces from old locally-bound desktops to every possible place. Mobile devices like smartphones, tablet-computers, notebooks, etc. allow us to work wherever we are, completely unbound from any location. P2P overlays, especially those which offer a location-based service are affected by the mobility of these devices. Other P2P applications are indirectly affected by the movement of its users. In some cases, or better to say at some places, a user might for example not want to reveal his or her identity in the network as it could be open to every other user. Whereas, in other cases a user wants to exhibit his/her position to other persons in the network in order to provide optimized routing strategies in this area.

In order to deepen the knowledge about overlay stacks and combinatorial P2P solutions, further costs and limitations of our proposed stack should be investigated. Probably, the placing of specialized single-purpose overlays on top of multi-purpose basic functionalities results in performance loss in comparison to optimized stand-alone overlay applications. Since one part of our solution is to combine different services dynamically during runtime, we expect the quality of different services to follow the dynamics of the overlay stack. Broader investigations of the quality of single services in the overlay stack under dynamical overlay changes are necessary. Once, metrics to analyze the quality of the overlay stack have been identified, the stack should be optimized, so that guarantees about different routing strategies can be given for different situations/states the overlay is in.

With the knowledge about the quality of services inside the overlay stack, adaptive publish-subscribe or multicast protocols as specialized overlay on top of the stacked overlay basis could be realized. The challenge is to adapt routing paths to sudden changes in the usage of the overlay stack. Another challenge is to efficiently use the base overlay and its integrated structures (and lightweight functionality) to forward and store streaming content within the network.

Routing over an unstructured friend-to-friend network structure for example guarantees a high degree of anonymity to forwarding nodes, since communication links between nodes can be created in a way that the nodes are required to consider each other to be friends. As a drawback, the time to search for (and to find) content in an unstructured network is often more costly than in structured DHT-based overlays in which a lookup is guaranteed to find a responsible peer for the query identifier. Possible future work could extend our overlay basis with a service which estimates the current quality of packet transmission, trying to adapt the routing behavior of participating nodes to find a suited route through the overlay.

We think this proposed idea is (loosely) related to the 3D tele-immersion project of the MONET group, lead by Klara Nahrstedt, University of Illinois. In specific, this future work might fit to the research on large scale real-time streaming.

### 9.2.2 A Location-based, Social, P2P-based Publish-Subscribe Service

The power of publish-subscribe (*pub-sub*) services lies in their ability to decouple consumers of information (*subscribers*) from the source (*publisher*) of the spread information. By this means, subscribers are able to receive messages/data without knowledge of its publisher. Publishers on the opposite do not have to know about their subscribers.

As part of the proposed overlay stack, existing P2P pub-sub mechanisms should be explored and integrated into our overlay stack. An endeavor which is challenging will be to integrate the pub-sub service into the existing overlay stack. Currently there exists only little work about stacked overlays so that it is not clear which effort is needed to integrate pub-sub functionalities into our proposed solution for a basic overlay.

Social and location-based component of our overlay can be used to implement the pub-sub service. The main idea is to use existing social relationships which are already included in our overlay stack as basis to disseminate information in the network. As part of a anonymity service, nodes are (additionally to structured links) connected in an unstructured way to their friends. Since friendship is often based on similar interests or a mutual friend, this unstructured links can be perfectly used to carry information to that point in the network where it is desired. Location-based structures on the opposite are utilized to spread information in a certain geographical area. News and other information concerning a specific region are systematically forwarded to subscribers located in this area.

The advantage of our proposed overlay stack is that structured and unstructured overlay functionalities can be used in parallel depending on the demanded performance of a user. While pub-sub services on top of a DHT promise to be reliable, the maintenance of the service might be costly. Pub-sub services built on top of randomly connected nodes, on the other hand, are easy to maintain, but might not guarantee all subscribers in a network to receive information reliably.

This project proposal might be related to the project *public/subscribe in dynamic networks* of the *Sonderforschungsbereich* 1053 at TU Darmstadt[4].

### 9.2.3 Mobile Crowd Sensing

In recent years, a growing presence of mobile devices could be observed, most of them are permanently or occasionally connected to the Internet. Most people stick to a permanently repeating movement pattern, their mobile devices follow their behavior. Mankind can be therefore seen as a big opportunistic network, each person is part of it and can be associated with a node in the network.

Most persons visit the same places day by day, only seldom they change their behavior. Also, rarely they change the way they use their devices at a specific location. A goal could be to extend our overlay stack by the functionality to treat the collection of (mobile) users as a big opportunistic network. Basic mechanisms of our stacked overlay could be extended to be aware of human movement patterns to be able to forward information due to existing patterns.

New contributions could be twofold, first, creating simulations of moving peers and their behavior in combination with our overlay stack. Second, extending the overlay stack by self-learning and adaptive mechanisms to adjust routing strategies to regular moves that are repeated. As result one might expect a delay-tolerant network in which persons are able to spread a query over multiple, opportunistically connected nodes. By doing this, the effect of communication islands (areas in the network which form a community that is not connected to the rest of the network) can be avoided, as different islands might be connected eventually during the meeting

[4] http://www.maki.tu-darmstadt.de/sfb_maki/ueber_maki/index.de.jsp

of to peers. Those loosely coupled ferries could be leveraged to carry data updates between separated networks. This would also allow to efficiently merge different network partitions.

### 9.2.4 Anonymous Peer-to-Peer Overlays

During our research, we identified many approaches for anonymous peer-to-peer overlays. Although all these approaches claim to be anonymous, there is typically no proof given for the anonymity of these overlays. Future work could focus on these overlays and analyze their anonymity. A possible metric to do so could be entropy as proposed in [23], [48], [124], and [49].

One general problem with anonymous peer-to-peer overlays is that structured overlays reveal too much information about their participants so that they are barely suited for anonymous communication. Anonymity is mostly reached through the mixing of communication paths using different relaying techniques. In the end, a peer receiving a message should not be able to reconstruct the path the message traversed nor should a peer be able to reconstruct the initiator of a message. Currently, unstructured peer-to-peer overlays seem to have best properties to support anonymity, although they are usually very slow in terms of data dissemination. Future work should therefore investigate the tradeoff between fast routing approaches and related anonymity.

## 9.3 Closing Words

Peer-to-peer systems are in the focus of research for almost two decades now. Many protocols which have been designed for the purpose of file sharing at first, evolved, at least in theory, to smart and distributed solutions for almost every possible use cases. It seems, as if every innovative network application has been realized as peer-to-peer variant. On the contrary, years before this thesis has been written, peer-to-peer was said to be *dead*. Even some well visited conferences which were completely dedicated to peer-to-peer topics died over the time. Rest in peace *IEEE International Conference on Peer-to-Peer Computing*, 2015 was the last year you have been alive.

Although many protocols and solutions for peer-to-peer systems and related issues exist, there are still open questions in this topic which remain unsolved. Most solutions in this field are supposed to support fast distribution of data at low costs, like it has been done in many file sharing applications. Other applications different from the typical file sharing approach, like Skype or Spotify have been realized as P2P systems. Nevertheless, most solutions are implemented as centralized client-server applications in the end, because server based applications allow better control over data, goods can be billed better, and malicious sources can be avoided more simply.

Other challenges coming with P2P applications is often the technical aspect. For end users who often need to configure their hardware to support P2P applications, the use of distributed tools is mostly no intuitive. The lack of common standards for P2P systems and the difficulty to test

P2P protocols is another barrier for development of new distributed applications. WebRTC is an open standard which emerges in the last years and seem to foster the realization and implementation of new protocols.

Aside from all technical challenges related to P2P applications, legal issues and ethical questions came up with the use of P2P systems, that are still not cleared. When the traffic of filesharing applications dominated the Internet, questions about copyright infringements came up, because P2P networks fostered the possibility to distribute data without being noticed, controlled, or blocked by a third party, for example a record company which was owner of the data. The motivation for this dissertation as a second example is that P2P applications for anonymous communication could be provided, especially in countries that literally repress the freedom of expression. In countries like China, Turkey, Egypt, but also the USA and others, surveillance of Internet traffic plays a great role. Many people in the world are monitored and traced, although they pretend to enjoy privacy and free speech. But in fact, many people are accused to be criminals just because they are assumed to be associated to specific topics. Various oficials and journalists in Turkey for example have been released from their duties or imprisoned because they are accused to be in contact with Fethullah Gülen, who is suspected by Recep Tayyip Erdogan the current president of Turkey, to be the initiator of the *coup d'état* in 2016. As can be seen in this example, in some countries of the world, the pronouncement of a certain ideology can cause personal problems which are possibly not justified.

Although distributed and anonymous communication can be used to hinder the surveillance, blockage, and control of information dissemination, as stated in this thesis, an ethical question arises with the use and existence of those tools: should we support tools, which allow anonymous and private communication to foster free speech and will on the one hand, where they could be abused by criminals on the other hand? Especially in these days, in which coward assassins are connected in a worldwide network, it is critical to offer anonymous communication tools which allow terrorists and other morons to spread their ideologies in an instant without being detected. Because of this fact, discussions about prohibitions of cryptography came up in many countries in the world. The dilemma is the following: on the one hand, P2P can help to preserve human rights that are threatened in different countries, on the other hand, it gives political hardliners that threaten human rights the possibility to communicate freely.

This might lead to another question which cannot be answered or discussed in this thesis: is it acceptable to fight terrorists, criminals, suppressive regimes, etc. which endanger human rights obviously in the same way and with the same weapons and violence, they apply? In other words: does someone who demands the observance of certain human rights have to obey them in a war against someone who does not obey and threatens them?

Few words in the end have to be said, even if they never reach the right audience: *be nice to each other*. Saying these words means to contribute to a peaceful ideology, even if it is a small amount. Also science, writing papers, and writing this thesis means to contribute a little part... even if P2P is said to be dead.

# Bibliography


[1] Osama Abboud, Aleksandra Kovacevic, Kalman Graffi, Konstantin Pussep, and Ralf Steinmetz. "Underlay Awareness in P2P Systems: Techniques and Challenges". In: *Proceedings of the International Symposium on Parallel and Distributed Processing (IPDPS)*. 2009 (Page: 55).

[2] Shahbaz Akhtar Abid, Mazliza Othman, and Nadir Shah. "3D P2P Overlay Over MANETs". In: *Computer Networks* 64 (2014), pp. 89–111 (Page: 47).

[3] Shahbaz Akhtar Abid, Mazliza Othman, and Nadir Shah. "A Survey on DHT-based Routing for Large-scale Mobile Ad Hoc Networks". In: *ACM Computing Surveys (CSUR)* 47.2 (2015), p. 20 (Page: 47).

[4] Shahbaz Akhtar Abid, Mazliza Othman, Nadir Shah, Osama Sabir, Mazhar Ali, Jawad Shafi, Saleem Ullah, et al. "Merging of DHT-based Logical Networks in MANETs". In: *Transactions on Emerging Telecommunications Technologies* 26.12 (2015), pp. 1347–1367 (Page: 47).

[5] Read Al-Aaridhi, Ahmet Yüksektepe, Tobias Amft, and Kalman Graffi. "Distributed Data Structures Improvement for Collective Retrieval Time". In: *Proceedings of the International Symposium on Wireless Personal Multimedia Communications (WPMC)*. 2016 (Page: 2).

[6] Tobias Amft. "Design, Implementation and Evaluation of Merging Mechanisms for Large-scale and Dynamically Partitioned Networks". Master's Thesis. Department of Computer Science, Heinrich Heine University Düsseldorf, 2014 (Page: 45).

[7] Tobias Amft and Kalman Graffi. *A Tale of Many Networks: Splitting and Merging of Chord-like Overlays in Partitioned Networks*. Technical Report: TR-2017-001. Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany, 2017 (Pages: 7, 13, 17, 45, 55, 61).

[8] Tobias Amft and Kalman Graffi. "Moving Peers in Distributed, Location-based Peer-to-Peer Overlays". In: *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*. 2017 (Pages: 6, 13, 17, 39, 42, 43, 55).

[9] Tobias Amft and Kalman Graffi. *The Benefit of Stacking Multiple Peer-to-Peer Overlays*. Technical Report: TR-2017-002. Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany, 2017 (Pages: 16, 17, 51, 53, 55, 56).

[10] Tobias Amft and Kalman Graffi. "Webs of Change? The Transformation of Online Social Networks and Communication Infrastructures from a Technological Point of View". In: *Academia in Transformation, Arab-German Young Academy of Sciences and Humanities (AGYA)*. 2016 (Pages: 11, 19).

[11] Tobias Amft, Barbara Guidi, Kalman Graffi, and Laura Ricci. "FRoDO: Friendly Routing over Dunbar-based Overlays". In: *Proceedings of the International Conference on Local Computer Networks (LCN)*. 2015 (Pages: 5, 13, 17, 33, 34, 37, 55).

[12] Filipe Araújo and Luís Rodrigues. "GeoPeer: A Location-aware Peer-to-Peer System". In: *Proceedings of the International Symposium on Network Computing and Applications (NCA)*. 2004 (Page: 41).

[13] Valerio Arnaboldi, Andrea Guazzini, and Andrea Passarella. "Egocentric Online Social Networks: Analysis of Key Features and Prediction of Tie Strength in Facebook". In: *Computer Communications* (2013) (Page: 29).

[14] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. "Analysis of Ego Network Structure in Online Social Networks". In: *Proceedings of the International Conference on Privacy, Security, Risk and Trust (PASSAT), and Proceedings of the International Conference on Social Computing (SocialCom)*. 2012 (Page: 29).

[15] Shah Asaduzzaman and Gregor von Bochmann. "GeoP2P: An Adaptive Peer-to-Peer Overlay for Efficient Search and Update of Spatial Information". In: *Computing Research Repository (CoRR)* abs/0903.3759 (2009) (Page: 41).

[16] Arsham Sabbaghi Asl. "Auswertung von parallel betriebenen, DHT-basierten Peer-to-Peer Overlays". Master's Thesis. Department of Computer Science, Heinrich Heine University Düsseldorf, 2017 (Pages: 14, 51, 55).

[17] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512 (Page: 34).

[18] Ingmar Baumgart, Bernhard Heep, and Stephan Krause. "OverSim: A Flexible Overlay Network Simulation Framework". In: *IEEE Global Internet Symposium*. 2007 (Page: 25).

[19] Stefan Behnel. "SLOSL - A Modelling Language for Topologies and Routing in Overlay Networks". In: *Proceedings of the International Conference on Parallel, Distributed and Network-based Processing (PDP)*. 2007 (Page: 54).

[20] Krista Bennett and Christian Grothoff. "GAP - Practical Anonymous Networking". In: *Proceedings of the International Workshop on Privacy Enhancing Technologies (PET)*. 2003 (Page: 35).

[21] Markus Benter, Mohammad Divband, Sebastian Kniesburges, Andreas Koutsopoulos, and Kalman Graffi. "Ca-Re-Chord: A Churn Resistant Self-stabilizing Chord Overlay Network". In: *Proceedings of the Annual Technical Conference on Networked Systems (NetSys)*. 2013 (Page: 47).

[22] Stevens Blond, Fabrice Fessant, and Erwan Merrer. "Finding Good Partners in Availability-aware P2P Networks". In: *Proceedings of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*. 2009 (Page: 29).

[23] Nikita Borisov. "Anonymous Routing in Structured Peer-to-Peer Overlays". PhD thesis. University of California, Berkeley, 2005 (Page: 66).

[24] Antoine Boutet, Anne-Marie Kermarrec, Erwan Le Merrer, and Alexandre Van Kempen. "On the Impact of Users Availability in OSNs". In: *Proceedings of the Workshop on Social Network Systems (SNS)*. 2012 (Page: 29).

[25] Giacomo Brambilla, Marco Picone, Michele Amoretti, and Francesco Zanichelli. "An Adaptive Peer-to-Peer Overlay Scheme for Location-based Services". In: *Proceedings of the International Conference on Network Computing and Applications (NCA)*. 2014 (Page: 41).

[26] Dan Bricklin. *Friend-to-Friend Networks*. http://www.bricklin.com/f2f.htm. Accessed: Feb. 2017 (Pages: 29, 35).

[27] Sonja Buchegger, Doris Schiöberg, Le-Hung Vu, and Anwitaman Datta. "PeerSoN: P2P Social Networking: Early Experiences and Insights". In: *Proceedings of the Workshop on Social Network Systems (SNS)*. 2009 (Page: 30).

[28] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. "MaxProp: Routing for Vehicle-based Disruption-tolerant Networks". In: *Proceedings of the International Conference on Computer Communications (INFOCOM)*. 2006 (Page: 47).

[29] Brendan Burns, Oliver Brock, and Brian Neil Levine. "MV Routing and Capacity Building in Disruption Tolerant Networks". In: *Proceedings of the International Conference on Computer Communications (INFOCOM)*. 2005 (Page: 47).

[30] David L Chaum. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". In: *Communications of the ACM* 24.2 (1981), pp. 84–90 (Page: 35).

[31] Yatin Chawathe, Sriram Ramabhadran, Sylvia Ratnasamy, Anthony LaMarca, Scott Shenker, and Joseph M. Hellerstein. "A Case Study in Building Layered DHT Applications". In: *Proceedings of the International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*. 2005 (Pages: 41, 54).

[32] Ahmad Cheraghi, Tobias Amft, Salem Sati, Philipp Hagemeister, and Kalman Graffi. "The State of Simulation Tools for P2P Networks on Mobile Ad-Hoc and Opportunistic Networks". In: *Proceedings of the International Conference on Computer Communication and Networks (ICCCN)*. 2016 (Pages: 11, 23–25).

[33] Chi-Yin Chow, Mohamed F. Mokbel, and Xuan Liu. "Spatial Cloaking for Anonymous Location-based Services in Mobile Peer-to-Peer Environments". In: *GeoInformatica* 15.2 (2011), pp. 351–380 (Page: 55).

[34] Ian Clarke, Oskar Sandberg, Matthew Toseland, and Vilhelm Verendel. "Private Communication Through a Network of Trusted Connections: The Dark Freenet". In: *Network* (2010) (Page: 35).

[35] Bram Cohen. "Incentives Build Robustness in BitTorrent". In: *Workshop on Economics of Peer-to-Peer systems*. 2003 (Pages: 2, 12).

[36] Marco Conti, Andrea Passarella, and Fabio Pezzoni. "A Model for the Generation of Social Network Graphs". In: *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*. 2011 (Page: 29).

[37] Marco Conti, Andrea De Salve, Barbara Guidi, and Laura Ricci. "Epidemic Diffusion of Social Updates in Dunbar-based DOSN". In: *European Conference on Parallel Processing (Euro-Par)*. Springer, 2014, pp. 311–322 (Page: 29).

[38] Marco Conti, Andrea De Salve, Barbara Guidi, Francesco Pitto, and Laura Ricci. "Trusted Dynamic Storage for Dunbar-based P2P Online Social Networks". In: *On the Move to Meaningful Internet Systems (OTM)*. 2014 (Page: 29).

[39] Leucio Antonio Cutillo, Refik Molva, and Melek Önen. "Safebook: A Distributed Privacy Preserving Online Social Network". In: *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*. 2011 (Page: 29).

[40] Frank Dabek, Ben Y. Zhao, Peter Druschel, John Kubiatowicz, and Ion Stoica. "Towards a Common API for Structured Peer-to-Peer Overlays". In: *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*. 2003 (Page: 54).

[41] Anwitaman Datta. "Merging Ring-structured Overlay Indices: Toward Network-data Transparency". In: *Computing* 94.8-10 (2012), pp. 783–809 (Page: 47).

[42] Anwitaman Datta and Karl Aberer. "The Challenges of Merging Two Similar Structured Overlays: A Tale of Two Networks". In: *Proceedings of the International Conference on Self-organizing Systems (SOS)*. 2006 (Page: 47).

[43] Anwitaman Datta and Rajesh Sharma. "GoDisco: Selective Gossip Based Dissemination of Information in Social Community Based Overlays". In: *Proceedings of the International Conference on Distributed Computing and Networking (ICDCN)*. 2011 (Page: 29).

[44] Anwitaman Datta, Sonja Buchegger, Le-Hung Vu, Thorsten Strufe, and Krzysztof Rzadca. "Decentralized Online Social Networks". In: *Handbook of Social Network Technologies and Applications*. 2010 (Page: 29).

[45] Andrea De Salve, Barbara Guidi, and Laura Ricci. "An Analysis of Ego Network Communities and Temporal a Affinity for Online Social Networks". In: *Proceedings of the International Conference on Smart Objects and Technologies for Social Good (GOODTECHS)*. 2016 (Page: 29).

[46] Andrea De Salve, Barbara Guidi, and Laura Ricci. "Evaluation of Structural and Temporal Properties of Ego Networks for Data Availability in DOSNs". In: *Mobile Networks and Applications* (2017), pp. 1–12 (Page: 29).

[47] Andrea De Salve, Marco Dondio, Barbara Guidi, and Laura Ricci. "The impact of user's availability on Online Ego Networks: a Facebook analysis". In: *Computer Communications* 73 (2016), pp. 211–218 (Page: 29).

[48] Yuxin Deng, Jun Pang, and Peng Wu. "Measuring Anonymity With Relative Entropy". In: *Proceedings of the International Workshop on Formal Aspects in Security and Trust (FAST)*. 2006 (Page: 66).

[49] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. "Towards Measuring Anonymity". In: *Proceedings of the International Workshop on Privacy Enhancing Technologies (PET)*. 2002 (Page: 66).

[50] Andreas Disterhöft and Kalman Graffi. "CapSearch: Capacity-Based Search in Highly Dynamic Peer-to-Peer Networks". In: *Proceedings of the International Conference on Advanced Information Networking and Applications (AINA)*. 2017 (Page: 3).

[51] Andreas Disterhöft and Kalman Graffi. "Convex Hull Watchdog: Mitigation of Malicious Nodes in Tree-Based P2P Monitoring Systems". In: *Proceedings of the International Conference on Local Computer Networks (LCN)*. 2016 (Page: 3).

[52] Andreas Disterhöft and Kalman Graffi. "Protected Chords in the Web: Secure P2P Framework for Decentralized Online Social Networks". In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2015 (Page: 3).

[53] Marco Dorigo. "Ant Colony Optimization". In: *Scholarpedia* 2.3 (2007), p. 1461 (Page: 35).

[54] Peter Druschel and Antony I. T. Rowstron. "PAST: A Large-scale, Persistent Peer-to-Peer Storage Utility". In: *Proceedings of the International Workshop on Hot Topics in Operating Systems (HotOS)*. 2001 (Page: 54).

[55] Robin I. M. Dunbar. "The Social Brain Hypothesis". In: *Evolutionary Anthropology: Issues, News, and Reviews* 6.5 (1998), pp. 178–190 (Pages: 12, 28, 29, 34).

[56] Nathan S. Evans, Chris GauthierDickey, and Christian Grothoff. "Routing in the Dark: Pitch Black". In: *Proceedings of the International Conference on Computer Security Applications (ACSAC)*. 2007 (Page: 35).

[57] Matthias Feldotto, Christian Scheideler, and Kalman Graffi. “HSkip+: A Self-stabilizing Overlay Network for Nodes with Heterogeneous Bandwidths”. In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2014 (Page: 47).

[58] Agostino Forestiero, Emilio Leonardi, Carlo Mastroianni, and Michela Meo. “Self-Chord: A Bio-inspired P2P Framework for Self-organizing Distributed Systems”. In: *IEEE/ACM Transactions on Networking* 18.5 (2010), pp. 1651–1664 (Page: 35).

[59] Jennifer Golbeck and Matthew Rothstein. “Linking Social Networks on the Web with FOAF: A Semantic Web Case Study”. In: *Proceedings of the International Conference on Artificial Intelligence (AAAI)*. 2008 (Pages: 29, 35).

[60] Bruno Gonçalves, Nicola Perra, and Alessandro Vespignani. “Modeling Users’ Activity on Twitter Networks: Validation of Dunbar’s Number”. In: *PLoS ONE* 6.8 (2011), e22656 (Page: 29).

[61] Kalman Graffi. “PeerfactSim. KOM: A P2P system simulator—Experiences and lessons learned”. In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2011 (Page: 25).

[62] Kalman Graffi, Sergey Podrajanski, Patrick Mukherjee, Aleksandra Kovacevic, and Ralf Steinmetz. “A Distributed Platform for Multimedia Communities”. In: *Proceedings of the International Symposium on Multimedia (ISM)*. 2008 (Pages: 30, 54).

[63] Kalman Graffi, Parag S. Mogre, Matthias Hollick, and Ralf Steinmetz. “Detection of Colluding Misbehaving Nodes in Mobile Ad Hoc and Wireless Mesh Networks”. In: *Proceedings of the International Conference on Global Telecommunications (GLOBECOM)*. 2007 (Page: 47).

[64] 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 International Conference on Peer-to-Peer Computing (P2P)*. 2010 (Pages: 30, 54).

[65] 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 International Conference on Consumer Communications and Networking (CCNC)*. 2011 (Pages: 30, 54).

[66] Kalman Graffi, Dominik Stingl, Julius Rueckert, Aleksandra Kovacevic, and Ralf Steinmetz. “Monitoring and Management of Structured Peer-to-Peer Systems”. In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2009 (Page: 54).

[67] Kalman Graffi, Patrick Mukherjee, Burkhard Menges, Daniel Hartung, Aleksandra Kovacevic, and Ralf Steinmetz. “Practical Security in P2P-based Social Networks”. In: *Proceedings of the International Conference on Local Computer Networks (LCN)*. 2009 (Pages: 30, 54).

[68] 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: *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS)*. 2008 (Page: 54).

[69] Kálmán György Graffi. “Monitoring and Management of Peer-to-peer Systems”. PhD thesis. Technische Universität Darmstadt, 2010 (Page: 3).

[70] Christian Gross, Dominik Stingl, Björn Richerzhagen, Andreas Hemel, Ralf Steinmetz, and David Hausheer. "Geodemlia: A Robust Peer-to-Peer Overlay Supporting Location-based Search". In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2012 (Pages: 18, 39, 41, 42, 61).

[71] Barbara Guidi. "P2P Architectures For Distributed Online Social Networks". In: *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS)*. 2013 (Page: 29).

[72] 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 (PPNA)* 9.6 (2016), pp. 1177–1194 (Pages: 2, 4, 12, 17, 27, 55).

[73] Tobias Heer, Stefan Götz, Simon Rieche, and Klaus Wehrle. "Adapting Distributed Hash Tables for Mobile Ad Hoc Networks." In: *Proceedings of the International Conference on Pervasive Computing and Communications (PerCom)*. 2006 (Page: 47).

[74] Thomas R Henderson, Mathieu Lacage, George F Riley, C Dowell, and J Kopena. "Network Simulations With the NS-3 Simulator". In: *SIGCOMM demonstration* 14.14 (2008), p. 527 (Page: 25).

[75] Dominic Heutelbeck and Matthias Hemmje. "A Peer-to-Peer Data Structure for Dynamic Location Data". In: *Proceedings of the International Conference on Pervasive Computing and Communications (PerCom)*. 2006 (Page: 41).

[76] Sushant Jain, Kevin R. Fall, and Rabin K. Patra. "Routing in a Delay Tolerant Network". In: *Proceedings of the International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*. 2004 (Page: 47).

[77] Jens Janiuk, Alexander Mäcker, and Kalman Graffi. "Secure Distributed Data Structures for Peer-to-Peer-based Social Networks". In: *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS)*. 2014 (Page: 30).

[78] Dilip Antony Joseph, Jayanthkumar Kannan, Ayumu Kubota, Karthik Lakshminarayanan, Ion Stoica, and Klaus Wehrle. "OCALA: An Architecture for Supporting Legacy Applications over Overlays". In: *Proceedings of the International Conference on Networked Systems Design and Implementation (NSDI)*. 2006 (Page: 54).

[79] Verena Kantere, Spiros Skiadopoulos, and Timos K. Sellis. "Storing and Indexing Spatial Data in P2P Systems". In: *IEEE Transactions on Knowledge Data Engineering* 21.2 (2009), pp. 287–300 (Page: 41).

[80] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. "The ONE Simulator for DTN Protocol Evaluation". In: *Proceedings of the International Conference on Simulation Tools and Techniques (SIMUTools)*. 2009 (Page: 25).

[81] Charles Edwin Killian, James W Anderson, Ryan Braud, Ranjit Jhala, and Amin M Vahdat. "Mace: language support for building distributed systems". In: 42.6 (2007), pp. 179–188 (Page: 54).

[82] Zoltán Lajos Kis and Róbert Szabó. "Chord-Zip: A Chord-Ring Merger Algorithm". In: *IEEE Communications Letters* 12.8 (2008), pp. 605–607 (Pages: 45, 47, 61).

[83] Zoltán Lajos Kis and Róbert Szabó. "Interconnected Chord-rings". In: *Network Protocols and Algorithms* (2010) (Page: 47).

[84] Sebastian Kniesburges, Andreas Koutsopoulos, and Christian Scheideler. "Re-Chord: a Self-stabilizing Chord Overlay Network". In: *Theory of Computing Systems* 55.3 (2014), pp. 591–612 (Page: 47).

[85] Mirko Knoll and Torben Weis. "Optimizing Locality for Self-organizing Context-based Systems". In: *Proceedings of the International Workshop on Self-organizing Systems (IWSOS)*. 2006 (Page: 41).

[86] Aleksandra Kovacevic, Nicolas Liebau, and Ralf Steinmetz. "Globase.KOM - A P2P Overlay for Fully Retrievable Location-based Search". In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2007 (Page: 41).

[87] Jinwon Lee, Hyonik Lee, Seungwoo Kang, Su Myeon Kim, and Junehwa Song. "CISS: An Efficient Object Clustering Framework for DHT-based Peer-to-Peer Applications". In: *Computer Networks* 51.4 (2007), pp. 1072–1094 (Pages: 41, 42).

[88] Jinyang Li and Frank Dabek. "F2F: Reliable Storage in Open Networks". In: *Proceedings of the International Workshop on Peer-To-Peer Systems (IPTPS)*. 2006 (Page: 29).

[89] Shen Lin, François Taïani, and Gordon Blair. "Exploiting Synergies Between Coexisting Overlays". In: *Proceedings of the International Conference on Distributed Applications and Interoperable Systems (IFIP)*. 2009 (Page: 54).

[90] Nuno Lopes and Carlos Baquero. "Implementing Range Queries with a Decentralized Balanced Tree over Distributed Hash Tables". In: *Proceedings of the International Conference on Network-based Information Systems (NBiS)*. 2007 (Page: 41).

[91] Harsha V. Madhyastha, Arun Venkataramani, Arvind Krishnamurthy, and Thomas E. Anderson. "Oasis: an Overlay-aware Network Stack". In: *Operating Systems Review* 40.1 (2006), pp. 41–48 (Page: 54).

[92] Balasubramaneyam Maniymaran, Marin Bertier, and Anne-Marie Kermarrec. "Build One, Get One Free: Leveraging The Coexistence of Multiple P2P Overlay Networks". In: *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*. 2007 (Pages: 52, 54).

[93] Sergio Marti, Prasanna Ganesan, and Hector Garcia-Molina. "SPROUT: P2P Routing With Social Networks". In: *Proceedings of the International Conference on Extending Database Technology (EDBT)*. 2004 (Page: 30).

[94] Petar Maymounkov and David Mazieres. "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric". In: *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*. 2002 (Page: 2).

[95] Giuliano Mega, Alberto Montresor, and Gian Pietro Picco. "On Churn and Communication Delays in Social Overlays". In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2012 (Page: 29).

[96] James W. Mickens and Brian D. Noble. "Exploiting Availability Prediction in Distributed Systems". In: *Proceedings of the International Symposium on Networked Systems Design and Implementation (NSDI)*. 2006 (Page: 29).

[97] Parag S. Mogre, Kalman Graffi, Matthias Hollick, and Ralf Steinmetz. "A Security Framework for Wireless Mesh Networks". In: *Wireless Communications and Mobile Computing Special Issue: Architectures and Protocols for Wireless Mesh, Ad Hoc, and Sensor Networks* 11.3 (2010), pp. 371–391 (Page: 47).

[98] Parag S. Mogre, Kalman Graffi, Matthias Hollick, and Ralf Steinmetz. “AntSec, WatchAnt, and AntRep: Innovative Security Mechanisms for Wireless Mesh Networks”. In: *Proceedings of the International Conference on Local Computer Networks (LCN)*. 2007 (Page: 47).

[99] Mirco Musolesi, Stephen Hailes, and Cecilia Mascolo. “Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks”. In: *Proceedings of the International Conference on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*. 2005 (Page: 47).

[100] Beomseok Nam and Alan Sussman. “DiST: Fully Decentralized Indexing for Querying Distributed Multidimensional Datasets”. In: *Proceedings of the International Symposium on Parallel and Distributed Processing (IPDPS)*. 2006 (Page: 41).

[101] Rammohan Narendula, Thanasis G. Papaioannou, and Karl Aberer. “A Decentralized Online Social Network with Efficient User-driven Replication”. In: *Proceedings of the International Conference on Social Computing (SocialCom)*. 2012 (Page: 29).

[102] 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 International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*. 2012 (Page: 30).

[103] *NS-3*. http://www.nsnam.org/. Accessed: Aug. 2017 (Page: 25).

[104] *OMNeT++ Discrete Event Simulator*. https://omnetpp.org/. Accessed: Aug. 2017 (Page: 25).

[105] *OverSim: The Overlay Simulations Framework*. http://www.oversim.org/. Accessed: Aug. 2017 (Page: 25).

[106] Lluis Pamies-Juarez, Pedro García López, and Marc Sánchez Artigas. “Heterogeneity-aware Erasure Codes for Peer-to-Peer Storage Systems”. In: *Proceedings of the International Conference on Parallel Processing (ICPP)*. 2009 (Page: 30).

[107] Thomas Paul, Benjamin Greschbach, Sonja Buchegger, and Thorsten Strufe. “Exploring Decentralization Dimensions of Social Networking Services: Adversaries and Availability”. In: *Proceedings of the International Workshop on Hot Topics on Interdisciplinary Social Networks Research (HotSocial)*. 2012 (Page: 29).

[108] *PeerfactSim.KOM - Community Edition*. http://www.peerfact.com/. Accessed: Aug. 2017 (Page: 25).

[109] A Pethalakshmi and C Jeyabharathi. “Geo-Chord: Geographical Location-based Chord Protocol in Grid Computing”. In: *International Journal of Computer Applications* 94.3 (2014) (Page: 41).

[110] Marco Picone, Michele Amoretti, and Francesco Zanichelli. “GeoKad: A P2P Distributed Localization Protocol”. In: *Proceedings of the International Conference on Pervasive Computing and Communications (PerCom)*. 2010 (Page: 41).

[111] Marco Picone, Michele Amoretti, and Francesco Zanichelli. “Proactive Neighbor Localization Based on Distributed Geographic Table”. In: *International Journal of Pervasive Computing and Communications* 7.3 (2011), pp. 240–263 (Page: 41).

[112] Bogdan C. Popescu, Bruno Crispo, and Andrew S. Tanenbaum. “Safe and Private Data Sharing with Turtle: Friends Team-Up and Beat the System”. In: *Proceedings of the International Workshop on Security Protocols*. 2004 (Page: 35).

[113] Vitaliy Rapp and Kalman Graffi. "Continuous Gossip-based Aggregation through Dynamic Information Aging". In: *Proceedings of the International Conference on Computer Communications and Networks (ICCCN)*. 2013 (Page: 54).

[114] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard M. Karp, and Scott Shenker. "A Scalable Content-addressable Network". In: *Proceedings of the International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*. 2001 (Page: 41).

[115] Sean C. Rhea, Brighten Godfrey, Brad Karp, John Kubiatowicz, Sylvia Ratnasamy, Scott Shenker, Ion Stoica, and Harlan Yu. "OpenDHT: A Public DHT Service and its Uses". In: *Proceedings of the International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*. 2005 (Page: 30).

[116] Sam G.B. Roberts, Robin I.M. Dunbar, Thomas V. Pollet, and Toon Kuppens. "Exploring Variation in Active Network Size: Constraints and Ego Characteristics". In: *Social Networks* 31.2 (2009), pp. 138–146 (Page: 29).

[117] Adolfo Rodriguez, Charles Edwin Killian, Sooraj Bhat, Dejan Kostic, and Amin Vahdat. "MACEDON: Methodology for Automatically Creating, Evaluating, and Designing Overlay Networks." In: *Proceedings of the International Conference on Networked Systems Design and Implementation (NSDI)*. 2004 (Page: 54).

[118] Antony I. T. Rowstron and Peter Druschel. "Pastry: Scalable, Decentralized Object Location, and Routing for Large-scale Peer-to-Peer Systems". In: *Proceedings of the International Conference on Distributed Systems Platforms*. 2001 (Pages: 34, 54, 60, 61).

[119] Antony I. T. Rowstron, Anne-Marie Kermarrec, Miguel Castro, and Peter Druschel. "SCRIBE: The Design of a Large-scale Event Notification Infrastructure". In: *Proceedings of the International Workshop on Networked Group Communication (COST264)*. 2001 (Page: 54).

[120] Oskar Sandberg. "Distributed Routing in Small-World Networks". In: *Proceedings of the International Workshop on Algorithm Engineering and Experiments (ALENEX)*. 2006 (Page: 35).

[121] Stefan Saroiu, Krishna P Gummadi, and Steven D Gribble. "Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts". In: *Multimedia Systems* 9.2 (2003), pp. 170–184 (Page: 2).

[122] Doris Schiöberg, Fabian Schneider, Gilles Trédan, Steve Uhlig, and Anja Feldmann. "Revisiting Content Availability in Distributed Online Social Networks". In: *Computing Research Repository (CoRR)* abs/1210.1394 (2012) (Page: 29).

[123] Mario T. Schlosser, Michael Sintek, Stefan Decker, and Wolfgang Nejdl. "HyperCuP - Hypercubes, Ontologies, and Efficient Search on Peer-to-Peer Networks". In: *Proceedings of the International Workshop on Agents and Peer-to-Peer Computing (AP2PC)*. 2002 (Page: 41).

[124] Andrei Serjantov and George Danezis. "Towards an Information Theoretic Metric for Anonymity". In: *Proceedings of the International Workshop on Privacy Enhancing Technologies (PET)*. 2002 (Page: 66).

[125] Tallat M Shafaat, Ali Ghodsi, and Seif Haridi. "Dealing With Network Partitions in Structured Overlay Networks". In: *Peer-to-Peer Networking and Applications* 2.4 (2009), pp. 334–347 (Pages: 45, 47, 61).

[126] Nadir Shah and Depei Qian. "Cross-layer Design to Merge Structured P2P Networks Over MANET". In: *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS)*. 2010 (Page: 47).

[127] Nadir Shah and Depei Qian. "Cross-layer Design to Merge Unstructured P2P Networks Over MANET". In: *Proceedings of the International Conference on Ubiquitous Information Technologies and Applications (CUTE)*. 2010 (Page: 47).

[128] Ayman Shaker and Douglas S. Reeves. "Self-stabilizing Structured Ring Topology P2P Systems". In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2005 (Page: 47).

[129] Rajesh Sharma and Anwitaman Datta. "SuperNova: Super-peers based Architecture for Decentralized Online Social Networks". In: *Proceedings of the International Conference on Communication Systems and Networks (COMSNETS)*. 2012 (Page: 30).

[130] Rajesh Sharma, Anwitaman Datta, Matteo Dell'Amico, and Pietro Michiardi. "An Empirical Study of Availability in Friend-to-Friend Storage Systems". In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2011 (Page: 29).

[131] Wei Song, Ruixuan Li, Zhengding Lu, and Guangcan Yu. "FAN: A Scalable Flabellate P2P Overlay Supporting Multi-dimensional Attributes". In: *Proceedings of the International Conference on Advanced Information Networking and Applications (AINA)*. 2008 (Page: 41).

[132] Ekaterina Stepanova. "The Role of Information Communication Technologies In the 'Arab Spring'". In: *Implications beyond the Region. Washington, DC: George Washington University, PONARS Eurasia Policy Memo* 159 (2011), pp. 1–6 (Page: 6).

[133] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications". In: *Proceedings of the International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*. 2001 (Pages: 5, 34, 60, 61).

[134] Alistair Sutcliffe, Robin Dunbar, Jens Binder, and Holly Arrow. "Relationships and the Social Brain: Integrating Psychological and Evolutionary Perspectives". In: *British Journal of Psychology* 103.2 (2012), pp. 149–168. ISSN: 2044-8295. DOI: 10.1111/j.2044-8295.2011.02061.x (Page: 29).

[135] Yuzhe Tang, Shuigeng Zhou, and Jianliang Xu. "LIGHT: A Query-efficient Yet Low-maintenance Indexing Scheme over DHTs". In: *IEEE Transactions on Knowledge Data Engineering* 22.1 (2010), pp. 59–75 (Page: 41).

[136] Egemen Tanin, Aaron Harwood, and Hanan Samet. "Using a Distributed Quadtree Index in Peer-to-Peer Networks". In: *VLDB Journal* 16.2 (2007), pp. 165–178 (Page: 41).

[137] *The ONE*. https://www.netlab.tkk.fi/tutkimus/dtn/theone/. Accessed: Aug. 2017 (Page: 25).

[138] Raúl Gracia Tinedo, Marc Sánchez Artigas, and Pedro García López. "Analysis of Data Availability in F2F Storage Systems: When Correlations Matter". In: *Proceedings of the International Conference on Peer-to-Peer Computing (P2P)*. 2012 (Page: 29).

[139] Jon Udell, Nimisha Asthagiri, and Walter Tuvell. "Security". In: *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. 2001, pp. 222–238 (Page: 35).

[140] András Varga and Rudolf Hornig. "An Overview of the OMNeT++ Simulation Environment". In: *Proceedings of the International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*. 2008 (Page: 25).

[141] Duncan J Watts and Steven H Strogatz. "Collective Dynamics of 'Small-World' Networks". In: *Nature* 393.6684 (1998), pp. 440–442 (Pages: 34, 35).

[142] Philipp Wette and Kalman Graffi. "Adding Capacity-aware Storage Indirection to Homogeneous Distributed Hash Tables". In: *Proceedings of the Annual Technical Conference on Networked Systems (NetSys)*. 2013 (Page: 54).

[143] David Isaac Wolinsky, Pierre St. Juste, P. Oscar Boykin, and Renato J. O. Figueiredo. "OverSoc: Social Profile Based Overlays". In: *Proceedings of the International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE)*. 2010 (Page: 30).

[144] Thomas Zahn and Jochen H. Schiller. "DHT-based Unicast for Mobile Ad Hoc Networks". In: *Proceedings of the International Conference on Pervasive Computing and Communications Workshops (PerCom)*. 2006 (Page: 41).

[145] Changxi Zheng, Guobin Shen, Shipeng Li, and Scott Shenker. "Distributed Segment Tree: Support of Range Query and Cover Query over DHT". In: *Proceedings of the International Workshop on Peer-To-Peer Systems (IPTPS)*. 2006 (Page: 41).

# Tobias Amft

## Curriculum Vitae

## I Personal Details

| | |
|---|---|
| Name | Tobias Amft |
| Academic degree | M.Sc. |
| Birth: date / place | 29.02.1988 in Tönisvorst, Germany |
| Nationality | German |
| Family status | Married to Vanessa Amft, née Fausten |
| Personal interests | Sports, Music, Science |
| Languages | German, English, Latin |

### Address


| | |
|---|---|
| Office address | Heinrich Heine University in Düsseldorf |
| | Institute of Computer Science |
| | Technology of Social Networks |
| | Universitätsstr. 1, 40225 Düsseldorf |
| | Tel. 0211/8112047, Fax. 0211/8111638 |
| | amft@cs.uni-duesseldorf.de |
| | http://tsn.hhu.de |

### Academic and professional experience

| | |
|---|---|
| Since 04/2014 | Researcher in the field of peer-to-peer overlays at the Technology of Social Networks Lab, lead by Jun.-Prof. Dr.-Ing. Kalman Graffi, Heinrich Heine University Düsseldorf |
| Since 10/2009 | Studies of Physics, Heinrich Heine University Düsseldorf<br>Target degree: M.Sc. (Diplom equivalent) |
| 03/2014 | Master thesis: "Design, Implementation and Evaluation of Merging Mechanisms for Large-scale and Dynamically Partitioned Networks"<br>Supervisors: Jun.-Prof. Dr.-Ing. Kalman Graffi, Prof. Dr. Michael Schöttner<br>Advisor: Jun.-Prof. Dr.-Ing. Kalman Graffi<br>Grade: Very Good (1.0) |
| 06/2012 - 03/2014 | Studies of Computer Science, Heinrich Heine University Düsseldorf<br>Degree: M.Sc. (Diplom equivalent)<br>Focus of studies: computer networks, peer-to-peer systems, network security<br>Final Grade: Very good (1.0) |
| 06/2012 | Bachelor thesis: "Eine zustandsbasierte Experimentsteuerung über einen fehleranfälligen Kommunikationskanal"<br>Supervisors: Prof. Dr. Martin Mauve, Prof. Dr. Michael Schöttner<br>Advisor: Norbert Goebel, M.Sc.<br>Grade: Very Good (1.3) |
| 10/2008 - 06/2012 | Studies of Computer Science, Heinrich Heine University Düsseldorf<br>Degree: B.Sc.<br>Focus of studies: computer networks, image processing<br>Final Grade: Good (2.0) |
| 08/1994 – 06/2007 | Elementary school, secondary school and high school<br>Abitur Degree: "Allgemeine Hochschulreife"<br>Grade: Very good (1.0) |

### Memberships

| | |
|---|---|
| IEEE | Institute of Electrical and Electronics Engineers |

## II Research projects

### Research projects involvement

| | |
|---|---|
| OverConverge<br>04/2014 - 03/2016 | HHU-SSF-OverConverge, Heinrich Heine University Düsseldorf<br>Project: "Convergence of Hierarchical Overlay Networks (OverConverge)".<br>Funded by the Strategic Research Fund ("Strategischer Forschungsfonds") of the Heinrich Heine University Düsseldorf<br>Main tasks: research |

# III Leadership experience

## Supervision of students

9 supervised Master theses
11 supervised Bachelor theses
11 supervised student research projects

## Exhibitions

| | |
|---|---|
| Hannover Messe 2015 | Software: LibreSocial, Opptain, and WebP2P – University of Düsseldorf |
| Hannover Messe 2014 | Software for secure and decentralized social networks – University of Düsseldorf |

# IV Services for the community

## Engagement in the university

| | |
|---|---|
| 2017 | Refreshment of course: “first aid at work” |
| Since 2016 | Member of the ESAG-Band “*Schweinemensakapelle*” |
| Since 2015 | First aid personnel after attending course: “first aid at work” |
| Since 2014 | Developer for PeerfactSim.KOM, community edition at the Heinrich Heine University |
| 2014 | Planning committee: graduation ceremony at the Heinrich Heine University of Düsseldorf, primary manager for beverages |

## Reviews and joint reviews for journals

Wiley, International Journal of Communication Systems, 2017
Inderscience Publishers IJAHUC, International Journal of Ad Hoc and Ubiquitous Computing, 2015
Springer JSS, Journal of Systems and Software, 2015
IEEE COMST, Transactions on Communications Surveys and Tutorials, 2014
IEEE TCC, Transactions on Cloud Computing, 2014
Springer PPNA, Peer-to-Peer Networking and Applications, 2014

## Reviews and joint reviews for conferences

IEEE CCNC, Consumer Communications and Networking Conference, 2015
IEEE CCNC, Consumer Communications and Networking Conference, 2014
IEEE LSDVE, International Workshop on Large Scale Distributed Virtual Environments, 2014
IEEE APWiMob, Asia Pacific Conference on Wireless and Mobile, 2014

# V Teaching

### Lectures and exercises

| | |
|---|---|
| SS17 | Seminar "Opportunistic and Peer-to-Peer Networks", Heinrich Heine University Düsseldorf<br>Topics:introduction into peer-to-peer overlays<br>Limited responsibility: partial lecture presentation |
| SS16 | Lecture "Peer-to-Peer Systems", Heinrich Heine University Düsseldorf<br>Topics: structured P2P overlays, network simulator PeerfactSim.KOM<br>Limited responsibility: partial lecture presentation, introduction to simulator PeerfactSim.KOM |
| WS15/16 | Lecture "Computer Networks", Heinrich Heine University Düsseldorf<br>Topics: Basic topics in the field computer networks, especially application and transport layer of the OSI model,<br>Limited responsibility: exam and exercise preparation, supervision of programming assignment: "Implementation of a Webserver" |
| WS14/15 | Lecture "Peer-to-Peer Systems", Heinrich Heine University Düsseldorf<br>Topics: structured P2P overlays, network simulator PeerfactSim.KOM<br>Limited responsibility: partial lecture presentation, introduction to simulator PeerfactSim.KOM, organisation and presentation of two lectures |
| WS14/15 | Lecture "Computer Networks", Heinrich Heine University Düsseldorf<br>Topics: Basic topics in the field computer networks, especially application layer of the OSI model and network security<br>Limited responsibility: exam and exercise preparation, supervision of programming assignment: "Implementation of a Webserver" |
| SS14 | Lecture "Computer Science II", Heinrich Heine University Düsseldorf<br>Topics: Basic topics of computer organization and programming<br>Limited responsibility: exam preparation, supervision of programming assignment: "Implementation of a Shell emulator" |

### Seminars and lab exercises

| | |
|---|---|
| SS17 | Project Seminar "Opportunistic and Peer-to-Peer Networks"<br>Topic: Location-based and motion-aware Overlays, general replication mechanisms<br>Limited responsibility: topic conception, student supervision, project management, introduction to simulator PeerfactSim.KOM |
| SS15 | Project Seminar "Opportunistic and Peer-to-Peer Networks"<br>Topic: Location-based Overlay by using Space-filling Curves<br>Limited responsibility: topic conception, student supervision, project management, introduction to simulator PeerfactSim.KOM |

### Supervised master theses

1. Daniel Sathees Elmo: "Implementation and Evaluations of a Mapping-based and Location-aware Peer-to-Peer Overlay as Overlay Stack Extension", 2017

2. Kolja Salewski: “HierARCHy: a Hierarchical, Attribute-based Peer-to-Peer Network Using Recursive, Changing Hyperrectangles”, 2017
3. Arsham (Pakan) Sabbaghi Asl: “Analysis of DHT-based Peer-to-Peer Overlays Executed in Parallel”, 2017
4. Oktay Sarier: “Data Availability in P2P Overlays in the Presence of Merging Processes”, 2017
5. Niklas Först: “Design, Implementation and Evaluation of a Modular Distributed Online Social Network”, 2017
6. Timm Kenfenheuer: “Design and Implementation of an Anonymization Layer for Live Communication”, 2017
7. Stefan Schmid: “Simulation and Prediction of User-behavior in Peer-to-Peer Networks”, 2017
8. Andrej Morlang: “Measuring Anonymity in an Extended Ant-based Peer-to-Peer Overlay”, 2016
9. Ivaylo Radev: “Social Peer-to-Peer Overlays and Location-based Overlays”, 2016

## Supervised bachelor theses

1. Olga Batiukova: “Modular Implementation and Evaluation of a Location-aware Peer-to-Peer Overlay”, 2017
2. Patrick Szewior: “Interconnection of Multiple Network Models for Highly Scalable Network Simulations”, 2016
3. Marjan Basic: “Implementation and Evaluation of Dunbar-based Social Network Techniques”, 2015
4. Timm Kenfenheuer: “Implementation and Evaluation of a Freenet-based Protocol for Anonymization Purpose in Peer-to-Peer Networks”, 2015
5. Mustafa Yilmaz: “Implementation of a Mesh-based Overlay for Location-based Search”, 2015
6. Erol Yildrim: “Investigation of Anonymization Strategies in DHT-based Overlays”, 2015
7. Tobias Korfmacher: “Implementation and Evaluation of a Mobility Model in a Peer-to-Peer Simulator”, 2014
8. Ilham Amara: “Implementation and Evaluation of Network Partitioning Events and Merging Algorithms in Peer-to-Peer Overlays”, 2015
9. Andrej Morlang: “Design and Evaluation of an Anonymization Service for Structured P2P Overlays”, 2014
10. Kolja Salewski: “Extraction and Analysis of State Information in Peer-to-Peer Networks”, 2014
11. Stefan Schmid: “Implementation and Evaluation of a new Peer-to-Peer Module for the Network Simulators OMNeT++ and OverSim”, 2014

## Supervised student research projects

1. Tobias Korfmacher: “Investigation of Simulation Techniques for Mobile Networks and Their Implementation in PeerfactSim.KOM”, 2015 - 2017
2. Kolja Salewski: “Design of an Overlay Stack for Location-based and Underlay-aware Overlays”, 2015 - 2016
3. Daniel Sathees Elmo: “Investigation of Location-based and Opportunistic Peer-to-Peer Overlays”, 2015 - 2016
4. Arsham (Pakan) Sabbaghi Asl: “Merging Techniques and Algorithms in P2P Networks”, 2015 - 2016
5. Oktay Sarier: “Merging Techniques and Algorithms in P2P Networks”, 2015 - 2016
6. Niklas Först: “Modeling of Friendship Relationships and User Behavior for Simulation Purpose”, 2015 - 2016
7. Timm Kenfenheuer: “Investigation of Dunar-based Distributed Online Social Networks”, 2015 - 2016
8. Stefan Schmid: “Modeling of Friendship Relationships and User Behavior for Simulation Purpose”, 2015 - 2016
9. Andrej Morlang: “Anonymization Techniques in Social Peer-to-Peer Networks”, 2015 - 2016
10. Christoph Claßen: “Location-based Peer-to-Peer Networks”, 2014 - 2015
11. Ivaylo Radev: “Social Peer-to-Peer Overlays and Location-based Overlays”, 2014 - 2015

### Selected talks

| | |
|---|---|
| 14.11.2016 | "Distributed Data Structures Improvement for Collective Retrieval Time", Conference Talk, 19th International Symposium on Wireless Personal Multimedia Communications (WPMC), Shenzhen, China |
| 30.08.2016 | "Hide and Seek in the Web: Distributed and Anonymous Social Communication", Invited Talk, Sino-German-symposium-on-social-computing, Göttingen, Germany |
| 25.05.2016 | "Convergence of Hierarchical Overlay Networks (OverConverge)", Research Talk, Heinrich Heine University, Düsseldorf, Germany |
| 28.10.2015 | "FRoDO: Friendly Routing Over Dunbar-based Overlays", Conference Talk, 40th IEEE Conference on Local Computer Networks (LCN), Clearwater Beach, Florida, USA |
| 23.04.2015 | "Convergence of Hierarchical Overlay Networks (OverConverge)", Research Talk, Heinrich Heine University, Düsseldorf, Germany |
| 20.08.2014 | "Ring Reunion - Partitioning and Merging of P2P Networks", Research Talk, Heinrich Heine University, Düsseldorf, Germany |
| 09.05.2014 | "Möglichkeiten zur digitalen Kommunikation in Zeiten der totalen Überwachung", Invited Talk, Deutscher Gewerkschaftsbund (DGB), Düsseldorf, Germany |

## VI Publications

### Articles

[1] Barbara Guidi, Tobias Amft, Andrea De Salve, Kalman Graffi and Laura Ricci. DiDuSoNet: A P2P architecture for Distributed Dunbar Based Social Networks. *Springer Peer-to-Peer Networking and Applications, Special Issue of Springer Journal Peer-to-Peer Networking and Applications (PPNA)*, 2014.

### Reviewed conference papers

[1] Tobias Amft and Kalman Graffi. Moving Peers in Distributed, Location-based Peer-to-Peer Overlays. *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*, 2017.

[2] Ahmad Cheraghi, Tobias Amft, Salem Sati, Philipp Hagemeister, and Kalman Graffi The State of Simulation Tools for P2P Networks on Mobile Ad-Hoc and Opportunistic Networks. *Proceedings of the International Conference on Computer Communication and Networks (ICCCN)*, 2016.

[3] Tobias Amft, Barbara Guidi, Kalman Graffi, and Laura Ricci. FRoDO: Friendly Routing over Dunbar-based Overlays. *Proceedings of the International Conference on Local Computer Networks (LCN)*, 2015.

### Book chapters

[1] Tobias Amft and Kalman Graffi Webs of Change? The Transformation of Online Social Networks and Communication Infrastructures from a Technological Point of View. *Academia in Transformation, Arab-German Young Academy of Sciences and Humanities (AGYA)*, 2016.

### Technical reports

[1] Tobias Amft and Kalman Graffi. The Benefit of Stacking Multiple Peer-to-Peer Overlays. *Technical Report: TR-2017-002*. Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany. 2017

[2] Tobias Amft and Kalman Graffi. A Tale of many Networks: Splitting and Merging of Chord-like Overlays in Partitioned Networks. *Technical Report: TR-2017-001*. Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany. 2017

### Theses

[1] Tobias Amft. *Design, Implementation and Evaluation of Merging Mechanisms for Large-scale and Dynamically Partitioned Networks*. Master thesis, Heinrich Heine University Düsseldorf, Germany, 2014.

[2] Tobias Amft. *Eine zustandsbasierte Experimentsteuerung über einen fehleranfälligen Kommunikationskanal*. Bachelor thesis, Heinrich Heine University Düsseldorf, Germany, 2012.

# Personal Publications

## Articles

Barbara Guidi, Tobias Amft, Andrea De Salve, Kalman Graffi and Laura Ricci. "DiDuSoNet: A P2P architecture for Distributed Dunbar Based Social Networks". In: *Springer Peer-to-Peer Networking and Applications, Special Issue of Springer Journal Peer-to-Peer Networking and Applications (PPNA)*. 2014.

## Reviewed conference papers

Tobias Amft and Kalman Graffi. "Moving Peers in Distributed, Location-based Peer-to-Peer Overlays". In: *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*. 2017. Acceptance Rate: 29%

Ahmad Cheraghi, Tobias Amft, Salem Sati, Philipp Hagemeister, and Kalman Graffi. "The State of Simulation Tools for P2P Networks on Mobile Ad-Hoc and Opportunistic Networks". In: *Proceedings of the International Conference on Computer Communication and Networks (ICCCN)*. 2016. Acceptance Rate: 30%

Tobias Amft, Barbara Guidi, Kalman Graffi, and Laura Ricci. "FRoDO: Friendly Routing over Dunbar-based Overlays". In: *Proceedings of the International Conference on Local Computer Networks (LCN)*. 2015. Acceptance Rate: 30.3%

## Book chapters

Tobias Amft and Kalman Graffi. "Webs of Change? The Transformation of Online Social Networks and Communication Infrastructures from a Technological Point of View". In: *Academia in Transformation, Arab-German Young Academy of Sciences and Humanities (AGYA)*. 2016.

## Technical reports

Tobias Amft and Kalman Graffi. "The Benefit of Stacking Multiple Peer-to-Peer Overlays". *Technical Report: TR-2017-002.* Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany. 2017

Tobias Amft and Kalman Graffi. "A Tale of many Networks: Splitting and Merging of Chord-like Overlays in Partitioned Networks". *Technical Report: TR-2017-001.* Technology of Social Networks Group, Heinrich Heine University, Düsseldorf, Germany. 2017

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 | Tobias Amft |