HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Secure Distributed Data Structures in Peer-to-Peer Networks

Inaugural-Dissertation

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

vorgelegt von

Raed Al-Aaridhi

geboren in

Bagdad, Irak

Düsseldorf, September 2018

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. Kálmán Graffi
Korreferent: Prof. Dr. Michael Schöttner
Tag der mündlichen Prüfung: 26.11.2018

Abstract

Nowadays, the Internet is very important in human daily life and has the potential to change the society via a large scale of applications which play a great role to develop a wide range of fields such as in industry and to enable basic communication between wide ranges of the people. The majority of these applications are provided technically from single entities such as a server or group of servers as a centralized point of service. They inherently carry the risk of misuse of private information, sensitive data leakage, snooping and spying on the users due to the central access of the supplier. Peer-to-peer (P2P) networks emerged in the Internet in order to provide new network functionality to interested peers, such as direct user-to-user communication, without the need of central providers. Due to the features and advantages of the decentralized approach of P2P networks such as robustness, scalability and no single point of failure, various large-scale application domains with diverse functionality requirements may be addressed.

The users in P2P networks provide each other data and in order to build more advanced applications such as social networks or distributed computing graphs, more advanced data structures such as sets, lists and trees should be supported by a P2P storage overlay with convenient access to these entries. In this thesis, we focus on the improvement of storage approaches in P2P networks and therefore introduce a new concept for distributed data structures by designing a new storage layer for P2P networks which allow the storage of complex data structure, as well as guarantee that the data will still be available using a special replication mechanism. To achieve that, our approach is based on three main parts.

In the first part, we focus on storage data and this entails a suitable storage service layer in P2P networks. Therefore, we design and evaluate a new layer named Distributed Data Structure (DDS). This layer persists data on a distributed hash tables (DHT), among multiple users, providing advance data structures like lists, sets, trees or single objects. The main idea behind the implementation is that a set, list or tree is stored in the DHT as data items containing ID information, pointers, and payload. DDS distributes the storage of the data over various nodes. A DDS supports to store DDS, retrieve DDS, modify DDS, adding and deleting entries in a DDS. In the evaluation of this DDS concept, all elements of each DDS are successfully stored and retrieved. Further, the evaluation shows that the approach comes with low overhead and delay.

In the second part of the thesis, we deal with the challenges of adding services for this DDS storage layer, such as search mechanisms of metadata as well as computational elements to DDS. These services are essential for our DDS approach in order to add new functionality that is required by complex P2P applications. In detail, we propose an extension for the DDS called the Computational Data Element, which interprets and computes the DDS payload and thus supports distributed function resolution. Furthermore, we introduce efficient algorithms that support metadata searches for the DDS in structured P2P networks.

In the third part of the thesis, we deal with the security issues for our DDS approach. Therefore, we propose a secure model for DDS, which guarantees security and access control of the data storage in the DDS in the P2P network using security mechanisms. The proposed security mechanism works completely without any trust between nodes in the networks.

In conclusion, we propose a secure distributed data structure scheme, a new methodology to store advanced data structures such as sets, lists, and trees in P2P overlay networks. We show that sophisticated P2P applications such as online social networks can be built on top of the DDS scheme as well as that new functionalities can be added to our storage scheme on the top of DDS layer. In the evaluation, we show that the approach provides the desired functionality, distributed the load and comes with low overhead and delay. We further show that our approach is secure through its integrated security mechanisms which are proposed to guarantee that the DDS is secure. We believe that our secure DDS approach will benefit the creation of further P2P-based applications.

Zusammenfassung

Heutzutage ist das Internet sehr wichtig im täglichen Leben des Menschen und hat das Potenzial, die Gesellschaft durch eine Vielzahl von Anwendungen zu verändern, die eine große Rolle bei der Entwicklung eines breiten Spektrums spielen. Dieses Spektrum umfasst Bereiche aus beispielsweise der Industrie aber auch grundlegende Kommunikation zwischen weiten Teilen der Bevölkerung, welche es zu ermöglichen gilt. Die meisten dieser Anwendungen werden technisch von einzelnen Systemen wie einem Server oder einer Gruppe von Servern als zentrale Anlaufstelle bereitgestellt. Ein zentraler Zugang des Anbieters birgt das Risiko des Missbrauchs privater Informationen, des Verlustes sensibler Daten und des Ausspionierens der Nutzer. Peer-to-Peer-Netzwerke (P2PNetzwerke) sind im Internet entstanden, um interessierten Peers neue Netzwerkfunktionalitäten, wie z.B. direkte User-to-User-Kommunikation, ohne die Notwendigkeit zentraler Anbieter, zur Verfügung zu stellen. Aufgrund der Vorteile des dezentralen Ansatzes von P2Poverlays wie Skalierbarkeit, Robustheit und kein Single-Point-of-Failure können verschiedene große Anwendungsbereiche mit unterschiedlichen Funktionsanforderungen angesprochen werden.

Im einfachen Fall stellen Benutzer von P2P-Netzwerken einander Daten zur Verfügung. Komplexere Anwendungen, wie etwa soziale Netzwerke oder verteilte Computing-Graphen, bedürfen fortgeschrittene Datenstrukturen wie Mengen, Listen und Bäume, worauf durch ein komfortables P2P-Speicher-Overlay zugegriffen werden kann. In dieser Arbeit konzentrieren wir uns auf die Verbesserung von Speicheransätzen in P2P-Netzwerken und führen daher ein neues Konzept für verteilte Datenstrukturen ein, indem wir eine neue Speicherschicht für P2P-Netzwerke entwerfen, die die Speicherung komplexer Datenstrukturen ermöglicht und gewährleistet, dass die Daten weiterhin über einen speziellen Replikationsmechanismus verfügbar sind. Unser Ansatz besteht aus drei Hauptteilen, die für die Erreichung der Ziele notwendig sind.

Im ersten Teil konzentrieren wir uns auf Speicherdaten und dies beinhaltet eine geeignete Speicherserviceschicht in p2p-Netzwerken. Deshalb entwerfen und evaluieren wir eine neue Schicht namens Distributed Data Structure (DDS). Diese Schicht besteht aus Daten auf einer verteilten Hash-Tabelle (DHT), die auf mehrere Benutzer verteilt ist und erweiterte Datenstrukturen wie Mengen, Listen, Bäume oder einzelne Objekte bereitstellt. Die Hauptidee hinter der Implementierung ist, dass ein Set, eine Liste oder ein Baum im DHT als Datenelemente gespeichert wird, die ID-Informationen, Zeiger und Payload enthalten. DDS verteilt die Last der Datenspeicherung auf verschiedene Knoten. Das Verfahren unterstützt das Speichern von DDS, das Abrufen von DDS, das Ändern von DDS, das Hinzufügen und Löschen von Einträgen in einem DDS. Bei der Evaluation dieses DDS-Konzeptes werden alle Elemente jedes DDS erfolgreich gespeichert und abgerufen. Darüber hinaus zeigt die Evaluation, dass der Ansatz mit geringem Overhead und Verzögerungen verbunden ist.

Im zweiten Teil der Dissertation beschäftigen wir uns mit Herausforderungen des Hinzufügens von Diensten für diese DDS-Speicherschicht, wie z.B. Suchmechanismen von Metadaten sowie Berechnungselemente zu DDS. Diese Dienste sind für unseren DDS-Ansatz unerlässlich, um neue Funktionen hinzuzufügen, die von komplexen P2P-Anwendungen benötigt werden. Im Einzelnen schlagen wir eine Erweiterung für das DDS vor, genannt das Computational Data Element, das die DDS-Nutzlast interpretiert und berechnet und damit eine verteilte Funktionsauflösung unterstützt. Darüber hinaus führen wir effiziente Algorithmen ein, die die Metadatensuche für das DDS in strukturierten P2P-Netzwerken unterstützen.

Im dritten Teil der Arbeit beschäftigen wir uns mit den Sicherheitsfragen für unseren DDS-Ansatz. Wir stellen ein sicheres Modell für DDS vor, das die Sicherheit und Zugriffskontrolle der Datenspeicherung im DDS im p2p-Netzwerk durch gewisse Mechanismen gewährleistet. Dabei kommt der vorgeschlagene Sicherheitsmechanismus ohne Vertrauen zwischen den Teilnehmern des Netzwerks aus.

Zusammenfassend schlagen wir ein sicheres verteiltes Datenstrukturschema vor, welches eine neue Methode darstellt, um erweiterte Datenstrukturen wie Mengen, Listen und Bäume in P2P-Overlay-Netzwerken zu speichern. Wir zeigen, dass hochentwickelte P2P-Anwendungen wie soziale Online-Netzwerke auf dem DDS-Schema aufgebaut werden können und dass neue Funktionalitäten zu unserem Speichersystem hinzugefügt werden können. In der Evaluation zeigen wir, dass der Ansatz die gewünschte Funktionalität bereitstellt, die Last fair verteilt, lediglich einen geringen Overhead vorweist und geringe Verzögerung hinzufügt. Wir sind davon überzeugt, dass unser sicherer DDS-Ansatz der Entwicklung weiterer p2p-basierter Anwendungen zugute kommt.

Acknowledgments

In the name of Allah, the Beneficent, the Merciful. Praise be to Allah, Lord of the Worlds. The Beneficent, the Merciful. Owner of the Day of Judgment. Thee (alone) we worship; Thee (alone) we ask for help. Show us the straight path. The path of those whom Thou hast favoured. Not (the path) of those who earn Thine anger nor of those who go astray. (Al-Fatiha, the Opening chapter of the Holy Quran)

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis. I would like to thank my supervisor, Prof. Dr. Kalman Graffi, for his supervision, patience, immense knowledge, motivation, and constant support. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. Not forgotten, my appreciation to my co-supervisor, Prof. Dr. Michael Schöttner for his support and knowledge regarding this topic. I would like to thank all my friends in our research group at the HHU for their support. I would especially like to thank my friend Newton for his support through the final reviewing and improving the language.

I would like to thank my great family, brothers, sisters, and relatives for supporting me throughout the years, especially my parents. I am exceptionally grateful and dedicate this thesis to the soul of my father who was one of his wishes to see me with a Doctorate degree and to my mother. No words to describe my love for my parents. You were an example of kindness, compassion, and integrity. Thank you for showing and teaching me great values such as the happiness in helping others and not to live selfishly. Thank you for always being there for me. Without the two of you, I do not know where I would be. I realize I do not thank you enough for everything. And for that, I apologize. If I have learned anything while being away from you, it is that you are the most important people in my life, and I love you both more than anything. Thank you for asking Allah in your daily prayers that Allah to bless me and supporting me. Thank you for the daily call over the phone for the reassurance and support. Special appreciation goes to my wife Ruqaya Al-Zaidi for her endless support, for giving me strength and hope, for her patience, for her being amazing and of course for making me smile no matter how stressed I am. Finally, For my son Ahmed, I love to thank him for making me smile every day and he was the most important thing that I got during my Ph.D. study, and I would like to tell him I am sorry for the times I could not play with him during my thesis.

Contents

Chapter 1

Introduction

The Internet is a global network consisting of computers and machines that uses a special suite of protocols to facilitate communication referred to as the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. The Internet itself was founded in 1960 by Advanced Research Projects Agency's Wide Area Network (ARPANET) of the US Department of Defense (DoD). Later on, other US institutions and companies joined this network and the growth finally lead to the development of the World Wide Web (WWW) as we know it today. The Internet manifests the potential to impact the society in many ways, both positively and negatively, as it provides a platform for connecting people via large-scale applications such as online social networks and Internet of Things (IoT) applications. As of April 13, 2018, there was an estimated 3,893,634,846 Internet users worldwide[1] out of a possible world population[2] of 7,614,928,196 . This translates to 51.13% of the world's population as being active Internet users.

In order to implement an application on the internet, there are various options like using a server with a single operator or by using several servers and multiple operators, in these systems the devices are divided into two depending on the function of the devices, that is, servers that provide services and clients that use these services. These systems, however, have some disadvantage due to the centralized topology such as a single point of failure, censorship and central access by the provider who is able to block the Internet services, for example, the Egyptian government during Arab Spring.

An other option, where the functions are put in all of the clients and they provide each to other and this is termed peer-to-peer (P2P). P2P networks are based on the cooperation of devices to provide resources to each other, and every peer functions as both a server as well as a client. This network topology has been shown to be capable of overcoming the debilitating issues of centralization networks such as single point of failure or censorship.

In P2P networks, the users can carry the data and they are then self-responsible for storing and providing each other the content. In order to build more advance applications such as online social networks, distributed service-oriented, or for metadata structures in the field of compute centers and storage depending applications. It would be very professional if the data structure would be supported by the overlay or by the network, distributed data structure so data is very important in such applications and is fully distributed and decentralization.

[1] http://www.internetlivestats.com/
[2] http://www.worldometers.info/world-population/

1.1 Motivation and Problem Statements

Over the past three decades, role of the Internet in human interactions has increased almost exponentially, giving the Internet the *ability* to effect and affect society using the connections that exist between people via large-scale applications such as online social networks. Most services in the Internet depend on centralized networks. In the last two decades, decentralized networks such as P2P overlay networks have become popular and have emerged with the aim of overcoming the limitations of centralized networks.

One key motivation behind the development of P2P system was the need for scalability. It is an established fact that the performance of the server is seen to decrease with increase the number of clients in client-server networks. In P2P networks, on the other hand, performance of the network increases with increase in the number of the peers where more resources are provided. Another reason for developing P2P networks was to overcome the issue of a single point of failure. The clients in the centralized networks rely on a central access point to access needed services or resources. Furthermore, the single access point itself presents several security problems such as risk of information leakage, misuse of data by the provider and spying on the users because of the central access of the supplier.

P2P overlay networks are essentially logical layers that are implemented on the top of the existing network topology. P2P networks are further divided into two main categories, unstructured and structured P2P networks. Unstructured P2P networks are so named because there are no restrictions on the peers arrangement. The peers randomly connect to some subset of all peers. Examples of an unstructured network include: Napster [33] which is considered as one of the first applications built on P2P networks for sharing music and was later shut down due to legal issues; and Gnutella [33]. Structured P2P networks rely on some form of structure for peer/item placement. Most structured P2P networks rely on some overlay topology to manage the placement such as distributed hast tables (DHTs) where every peer or object maintains and store information as key-value pairs in DHT and every peer is identified by a unique id from a predefined identifier space. This allows the peers to undertake efficient retrieval using the associated key of the given peer/data item. Examples include such as in Chord [34], Pastry [31] and Kademlia [27]. For many reasons, the work established in thesis restricts itself to structured P2P which rely on the DHTs as they provide advantages such as self-organization, scalability and robustness against failure. Therefore application and services developed to test the P2P network are hinged heavily on the DHT.

In the past years, many research and solutions are introduced in the field of P2P and overlay networks, for enhancing the security properties for P2P networks as in [26, 13, 12, 35], for distributed monitoring such as in [24, 15, 20, 15], for routing and overlay modules such as in [8, 10, 11, 9] and many other research and applications are presented. Many of the current sophisticated applications which have established themselves as essential as global communication channel between people have considered the use of decentralized networks, for example, social networks like Safebook [14], and Lifesocial.KOM [22, 23, 19]. This has been because the centralized networks heavily rely on a single provider running service who can misuse content in different ways, and can also monitor and shut down the services to reduce or stop communications on specific topics. Examples of such blockage of communications include the Arab Spring, where the Egyptian government blocked the service; or ability to censor such as the NSA scandal; or the misuse of the privacy such as in Facebook–Cambridge Analytica data scandal.

The main limitation of using DHTs in such applications, especially for storage dependent applications, is the bottleneck of single object-oriented data storage. The DHTs support the storage and lookup of single data objects based on their identifiers (IDs) using put/get operations. This will suffice for simple applications such as file sharing applications. However, for the more advanced storage-dependent applications that will be storing large as well as complex data structures, this will be insufficient and impractical. Consider the following scenario for sophisticated application. A social network has an album set of a user. There are 30 albums for the user with each album storing 20 to 50 pictures. Furthermore, each picture may have a long list of comments. The sum total of this is approximately 1200-3000 individual data entries, which is similar to a single-object-based DHT. Hence, this single node (user/peer) might be overloaded and the replication scheme might take too long to replicate all 3000 data elements to other nodes. This points to the need to distribute the storage by designing a new storage layer within the P2P network which will then allow storage of this complex data structure with guarantee that the data will still be available using an appropriate replication mechanism.

The goal of this thesis is to introduce a secure distributed data structure for P2P networks by designing a new storage layer for P2P networks which allows to distribute the load by distributing the storage, and allow storage of complex data structure as well as to provide a guarantee that the data which will be stored in this scheme is kept secure. Our approach is threefold. First, we introduce distributed data structure (DDS) scheme to support complex and big data structures such as sets, lists, and trees as middleware service in P2P networks. Second, we develop and essential services to add new functionality of this DDS scheme such as metadata search mechanisms in DDS scheme. Third, based on this DDS scheme, we propose security and access control mechanisms for a flexible secure DDS in P2P networks.

In the following, we give an overview on the requirements in meeting the three research goals highlighted and also present the specific tasks needed to realize them in each case. For the secure distributed data structures in P2P networks, we aim at addressing the following requirements for a totally distributed and secure storage environment:

- **Storage Data:** This entails a suitable storage service layer in P2P overlay networks. This layer offers storage services that enables storage and retrieval of data in a distributed manner. The storage layer also guarantees data persistence and availability via a special replication protocol that is implemented.

- **Services:** These support DDS scheme by adding two essential services such as a metadata-based search mechanism as well as adding computational elements to DDS in order to add new functionality of DDS scheme.

- **Security:** Introduce secure model for DDS scheme, which guarantees security and access control of the data storage in the distributed data structure within the P2P network using security mechanisms. The proposed security mechanism works completely without any trust between nodes in the networks.

There are several challenges based on the problem statements that must be considered in the discussion of how to achieve the set goals. The first **Challenge 1**, focuses on data storage in

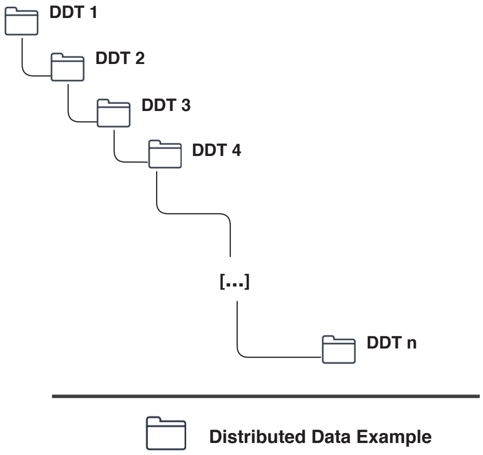


Figure 1.1: Deep DDT Structure with n Nodes in total.

P2P networks. During the investigation for future P2P-based, storage depending applications, one main limitation of structured P2P networks such as DHTs became apparent: the bottleneck of single-object oriented data storage. For future applications that need complex and large data structures, this single object focused view is a severe bottleneck, as we explained in the previous scenario, it becomes unpractical to store large data structures, such as a list of comments in article or albums of images in a single data object which lead to overloading. The questions that are derived from the problem statement described:

- How can storage of complex and big data structures in structured P2P networks be achieved so as to overcome the limitations seen in DHT that support the single data element storage as we explained in the previous scenario?
- How can distribution of the load be realized so that node overloading can be overcome?
- How can the success of retrieval operations of distributed data in P2P networks be guaranteed as well ensuring the data availability when nodes gone offline?

The second **Challenge 2**, concerns the time-consuming retrieval of deep data structures. In a deep tree structure as shown in Figure 1.1, the collective retrieval duration takes very long as there are many pointers to resolve iteratively. Each data item only resolves the pointer to the next element, leading to a chain of the individual item. With its deep reference chain, for each retrieved DDS object, a further DHT lookup is instructed. The resulting is increasing the duration times of collective-retrieves with the greater depth in DDS scheme due to the nature of DHTs. Therefore we desire a method which helps accelerate the retrieval process in the data structure. The questions derived from this problem statement described are:

- How can the retrieval operation in deep data structures be accelerated?

- How can parallelization of data retrieval be realized?

The third challenge named **Challenge 3**, considers adding new functionality of the storage scheme by the addition of services such as metadata-based searches in P2P networks, since the complex data structures addressed in **Challenge 1** will need suitable search services. P2P networks have no single query point because the data is stored among peers, while structured P2P networks provide a lookup mechanism rely on a simple key-value, and by this the lookup for a file or document with specific key in a fixed time could execute as shown to take O(log N) hops of lookup. Structured P2P networks lack standard algorithms to execute the metadata search. Metadata such as information or data is related to a specific file for example the name of author or title of book. Therefore, we need to introduce algorithms that support metadata searches in structured P2P networks as in DHT overlays. The questions derived from this problem statement described are:

- How can the search mechanisms in the storage scheme addressed in **Challenge 1** be added?

- Which searching methods are appropriate for use with DDS scheme?

The fourth challenge **Challenge 4**, looks at how to propose and evaluate new mechanisms for computational elements to ensure that the functions should be resolved in DDS scheme and the payload is up-to-date while ensuring correctness in P2P networks. Therefore, we need to integrate our DDS scheme with the computational elements which consist of an identifier, a function over time or a function over further data. The questions derived from this problem statement described are:

- How can the payload between the nodes each other in DDS scheme addressed in **Challenge 1** be exchanged since in most cases functions need value from other nodes?

- How can we implement the distributed methods in DDS scheme?

- What should be done in DDS scheme when there are two functions reference to each other, for example, A= (3+B) and B= (5+A), resulting in a loop? If this occurs in a computing system, the protocol would require both nodes to constantly update each other.

The fifth challenge **Challenge 5**, is the security of DDS scheme in P2P networks. Data security is very important because P2P networks are open networks which means every node, even malicious ones, can join and misuse the network in different ways such as stealing the data, spying or spreading viruses. Therefore, without proper security mechanisms, the idea of storing private and sensitive data on unknown network peers become pointless, as the stored data can be read or manipulated. Therefore, we need to present and evaluate a concept of a secure model for the DDS scheme which is proposed in **Challenge 1**. The aim is that such a solution works completely without trusted nodes for the DDS scheme. Therefore the secure model considers and investigates possible ways to handle the security problems, such as, access control, for the secure distributed data structure model in P2P networks which is proposed in **Challenge 1**. The questions derived from the security and access control problems statement described are:

- How can we design the security architecture for DDS scheme as a flexible model so that acceptable proportions for large P2P applications can be realized?
- How can we implement the essential features the security architecture should offer?
- How can a good access control mechanisms for secure DDS scheme in P2P networks that do not depend on trusted third parties be done?

The sixth and final challenge, which is named **Challenge 6**, is to have real sophisticated application which use the DDS scheme proposed in **Challenge 1**, therefore, we need to use DDS scheme in application such as Distributed Online Social Networks. Questions of concern are:

- How can we use the DDS scheme in real sophisticated applications?
- Which are the features such an application should offering?

As summary of this section, we discussed the research questions which are derived from the challenges described above. We aim to introduce a secure storage scheme as a storage layer in P2P networks supporting complex and big data structure such as set, list and trees with guarantee that the data still persists and is secure. Furthermore, this scheme is supported from services such as metadata search mechanisms and computational elements in order to add new functionality.

1.2 Contributions

The achievements for this dissertation are documented in eight publications broken in three main parts which form the succeeding three chapters. They highlight the achievements realized based on the goal of developing a secure distributed data structure scheme in P2P networks. First, we introduce the storage scheme offering distributed data storage services as a middleware in P2P networks, which guarantees distribution of data among the network participants, along with an appropriate replication technique. We called this scheme DDS. Peers that use this scheme can store and retrieve their data with high reliability. Thereafter, we focus on the services which support the DDS scheme, aiming at the improvement of the quality of DDS scheme and add new functionality such as search mechanisms for metadata. Finally, we examine the security and access control of storage data for DDS scheme in order to guarantee that data stored in an insecure system such as a P2P networks remains secure. We elaborate these points in the following short overview of contributions.

1.2.1 Distributed data structure scheme for P2P networks

To answer the questions in **Challenge 1**, we propose in [3] and in Chapter 2.1 a distributed storage scheme named DDS scheme which supports advanced data structures such as sets, lists, and trees which support sophisticated applications such as online social networks. Our

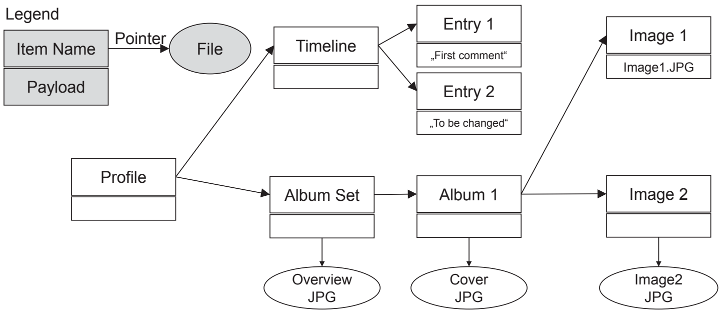


Figure 1.2: A Distributed Data Structures (DDS) in a simple scenario. © [2016] IEEE

approach is based on a distributed pointer-structure which is stored in the DHTs. DDS scheme is pointer graphs in the DHT, in which the linked data objects contain ID information, pointers and payload as shown in Figure 1.2. The DDS scheme supports individual operations on individual items, such as selective modification. In the evaluation, we show that all elements of each distributed data structure is successfully stored and retrieved and that the approach comes with

low overhead and delay. We have observed in a specific experiment when we use a deep tree structure as shown in Figure 1.1 that the collective retrieval duration takes quite long as there are many pointers to resolve iteratively as we mentioned in **Challenge 2**. As a solution for this problem, we propose in our paper [6] and in Chapter 2.2 a Structure Cache and save it along the root Distributed Data Trees (DDT) object of the DDS in order to solve the time-consuming retrieval of deep data structures which we present in [3]. The Structure Cache contains the overlay keys of all deeper distributed data objects under the cached root object. Upon retrieving the root DDT object, we gain the corresponding Structure Cache and all deeper DDT objects can be started to be retrieved simultaneously in parallel.

In the evaluation, we show that the Cache Structure gives us an alternative way to retrieve the complete DDS much faster with high reliability in the retrieval process. One limitation of the Structure Cache feature is the need to keep the Cache updated. Therefore, in order to overcome this limitation, we evaluate in our paper [1] and in Chapter 2.3 different approaches to update the Structure Caches which we present in [6]. The proposed approaches improve the performance of the Structure Cache which is used to make the retrieval operations of the deep DDS such as deep DDT much faster, and this leads to a significant enrichment for different applications such as in distributed social networks. The first updating approach for Structure Cache is named a Periodic Top-Down Update, using just the minimal Information about the structures that are used by the DDT anyways. While in the second updating approach, we introduce an Instant Bottom-Up Update approach, using additional information about the predecessors in the DDT to backtrack the DDT to its root, updating all caches on its way. In the evaluation, we show the advantages and disadvantages of both update mechanisms. Nonetheless, both could be the right choice for an application, depending on how an application uses the Structure Cache and what the main purpose of the application is.

1.2.2 Services for distributed data structures in P2P networks

DDS is a storage layer which uses and get benefits from the functions that have to offer in the under layers and at the same time, it should generate services in order to add new functionalists and improve the quality of the DDS scheme applications in P2P networks.

Therefore, in order to answer the questions in **Challenge 3**, we introduce in our paper [7] algorithms that support metadata searches in structured P2P networks as in DHTs in order to add new functionality to the DDS scheme which we present in our paper [3]. Metadata can generally be defined as data or information about a file or a document in the system, and it is important for the DDS scheme. The functionality of this DDS scheme would essentially be incomplete if an efficient mechanism for searching metadata is not considered. Therefore this work extends the DDS scheme proposed in our paper [3] by supporting metadata search algorithms to bolster the search functionality. In [7] and in Chapter 3.1, we propose four different search algorithms to perform the metadata search. In this case, the solutions differ based on how the join operation occurs and how the search process performs, so that, the proposed solutions are named Local Join, Parallel Join, Network Join and Bloom Join, which are successfully implemented on the DDS scheme. Finally, we discuss the results in order to show the advantages and disadvantages for each approach in the evaluations.

To answer the questions in **Challenge 4**, we introduce in [2] and in Chapter 3.2 an extension for the DDS scheme called Computational Data Elements (CDE) which interprets and computes the payload in the DDS scheme. A CDE is composed of an identifier, a function over time OR a function over further data. Computational elements are important for advanced applications, as they add new functionality to the DDS scheme [3] and allow for ease in extending the use cases of DDS scheme in distributed applications. They additionally ensure that the payload in the DDS scheme is up-to date with correct functions. As in the example where the elements are pictures of an album shown in Figure 1.3, we consider in this example the use case of the P2P-based online social networks which uses profiles, albums and therefore would benefit of the DDS scheme. In specific, we consider the Bob's profile who has a set of three albums, these albums store a different number of images as shown in Figure 1.3 Part B. In this example, the CDE extension will then require an additional backwards connection as highlighted in Figure 1.3 Part C, when the value of key A which represents the number of images, changes over time, for example, from 98 to 101, then the value of payload also should change, because of the payload is the sum of following arguments.

$$Payload = Value1 + Value2 + Value3 \tag{1.1}$$

Plus (+) is a function denoting "the sum up the following arguments". The sum is done by implementing four distribution methods called: push without timer, push with timer, pull without timer and pull with timer. Furthermore, the special case of many functions where each function is influenced by another function (chain of functions) is highlighted and discussed for example A= (5+B) and B= (2+A), resulting in a loop.

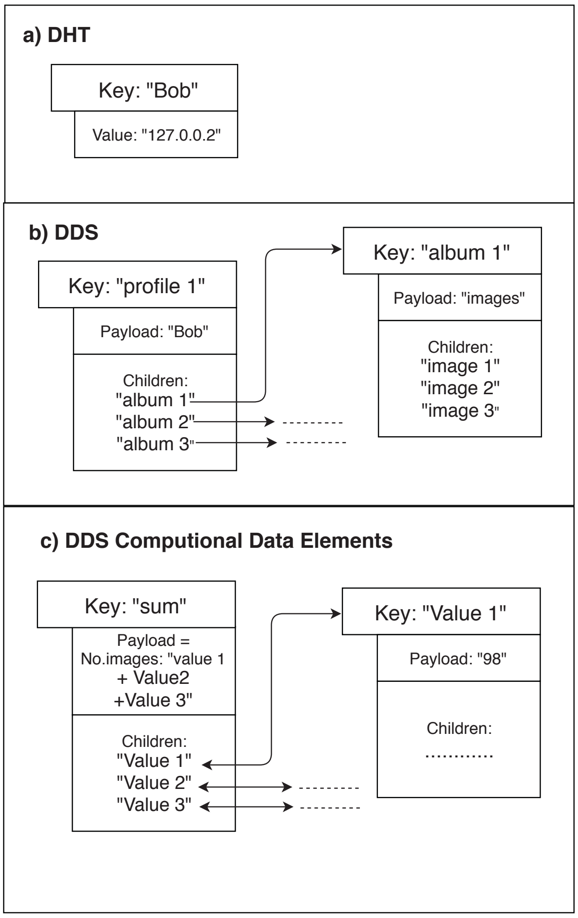


Figure 1.3: DDS schema with Computational Data Elements example.

1.2.3 Security and access control for distributed data structures in P2P networks

To answer the questions in **Challenge 5**, we present and evaluate a flexible security model for the DDS scheme which is presented in our paper [5] based on strictly defined security requirements in details. DDS scheme is implemented on the top of DHT, which consist of linked data items, within P2P networks. The data storage in the DDS scheme in our paper [3] was completely insecure even malicious nodes can join and cause havoc in the network. Therefore, in our paper [5] and in Chapter 4.1 we introduce a secure model which allows for storing of the DDS scheme within the P2P network securely. To achieve this goal different security mechanisms are implemented, such as hybrid encryption, granular encryption and encryption signature strategy. With these strategies, it is then possible to secure DDS scheme in [3] as a collection of multiple secured data items. In the evaluation, focus is on the measurement of the overhead of time and space caused by the DDS security model. The results show that using the security architecture within given feature set introduces inevitable linear time and space overhead while staying scalable overall for large P2P applications. Furthermore, in order to extend the security model in [5], we propose in [4] and in Chapter 4.2 an access control model for the DDS scheme which we present in [3]. For this purpose, different aspects of access control are discussed and initial assumptions concerning the security of the DDS scheme and its data environment are declared. We introduce an access control architecture for the DDS scheme working in DHTs without depending on trusted third parties. Many access problems are carefully analyzed such as the revocation problem of shared access to secured distributed system.

1.2.4 A sophisticated application using DDS scheme

One of the characteristics of the Distributed Online Social Network (DOSN) is that the participant's profile's data is managed by the peers that build up the DOSN. This leads to new security problems regarding the privacy and availability of participant data. The privacy of published content in the participant's profile in DOSNs is very important and should be protected because it contents private information and sensitive data. To answer the questions in Challenge 6, we propose in our paper [32] and in Chapter 4.3 a new approach to maintaining the participant's privacy and increase the participant's contents' availability without needing to encrypting the data encryption which is necessary to protect the confidentiality of participant $X'S$ data when stored on $Y'S$ peer. For that, our approach is composed of modeling the content belonging to the profile of participant X using a hierarchical tree data structure from the DDS scheme which we present in [3], and affiliate a suitable method for distribution the nodes of the tree on the peers who is currently online friends of X. The main idea of our approach

is to assign a copy of each data D of the tree to another peer who is presently online, and whose participant Y is allowed to access D depending on $X'S$ privacy policy. Therefore, with our approach we avoid the encryption which is necessary to protect the confidentiality of $X'S$ data when stored on $Y'S$ peer, because Y is a friend of X and X based on his privacy policy gave Y the right access to his data. In the evaluation, we consider five reference policies which depend on friendship relationships between the participants. The evaluation results showed that the proposed approach ensures the availability of the contents of the profiles significantly.

In Figure 1.4, we give a summary of the secure DDS scheme and the contributions of this thesis which are dealing with our objectives. The secure DDS scheme acts as a storage layer on top of a DHT overlay. In this scheme architecture, the bottom is the DHT layer, which has two important commands (Put and Get) for the storage and retrieval of data in P2P networks. On top of this layer, the DDS scheme offers the DDS replication service in order to replicate the data objects which are stored in the DHT to multiple DHT nodes, because nodes in the P2P networks go offline frequently, therefore, a replication service is essential for data availability in P2P networks. On the top of the replication service is the DDS layer which contains a different kind of the DDS such as DDTree, DDList and DDSet components which organize distributed data's with different constraints. DDS scheme consists of data items which contain an object ID under which this item is stored and optional pointers to further data items in the DDS graph as well as optional payload, such as an image. In order to add new functionality as well as to extend the use case of DDS, many essential extensions are added on the top of

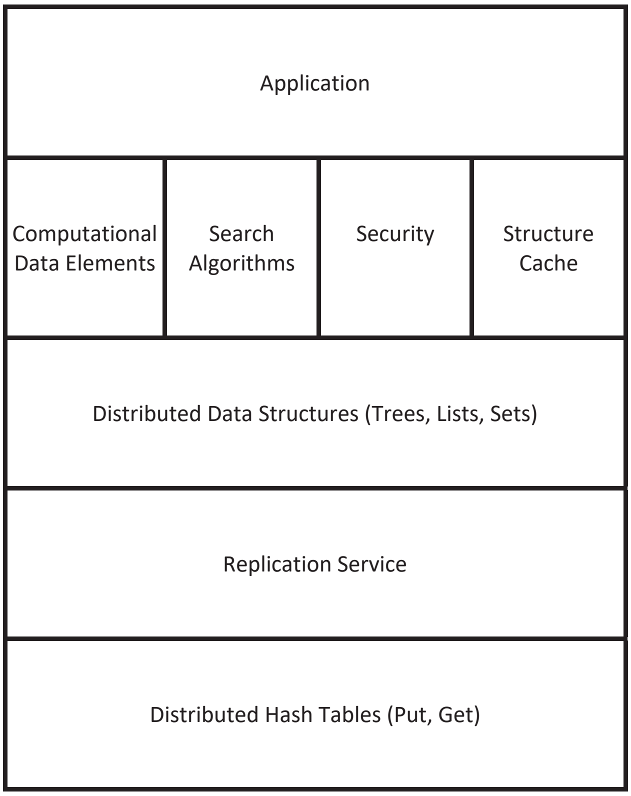

Application
Computational Data Elements
Search Algorithms
Security
Structure Cache
Distributed Data Structures (Trees, Lists, Sets)
Replication Service
Distributed Hash Tables (Put, Get)

Figure 1.4: A schematic representation gives the overview of the DDS scheme, as well as the contributions parts which are highlighted within this Thesis.

DDS layer, starting with Structure Cache which improves the performance of DDS scheme via accelerating the retrieval process in the deep DDS scheme such as the deep tree. In addition services such as search mechanisms of metadata as well as computational elements added to the DDS scheme in order to add new functionality. Furthermore, a secure model is added for the DDS scheme, which guarantees security and access control of the data storage in the DDS scheme using security mechanisms. Finally, the Application Layer which contains an example for advanced applications such as online social network.

1.3 Thesis Structure

This thesis is organized as the following sequence. Chapter 1 presents an introduction and motivation for the DDS scheme in P2P networks and its storage challenges and models. It lists the contributions and research questions that are discussed through this dissertation. Chapters 2-4 provide summaries for the published papers of the author which include 8 publications.

We present in Chapter 2 the challenges as well as our solutions for data storage in P2P networks, Chapter 2 includes the summaries for three publications which are distributed in three Sections, each one of them include one publication. In Section 2.1, we present the summary of [3], which proposes a distributed storage scheme named DDS scheme. In Section 2.2, we present the summary of [6], which motivates the Structure Cache which is used to accelerate the retrieval process in the DDS scheme, while in Section 2.3, we present the summary of [1], which evaluates different approaches to update the Structure Caches.

Chapter 3 includes a summary of two papers and discusses the essential services for DDS scheme. In Section 3.1 we present the summary of [7] which displays the search mechanisms for metadata for the DDS scheme in P2P networks, while in Section 3.2 we present the summary of [2] which includes the computational elements in P2P networks.

Chapter 4 presents a summary of three papers which is broken in 3 Sections, this Chapter discusses the security issues for DDS scheme in P2P networks and present our solutions. In Section 4.1 we present the summary of [5] in which we propose and evaluate a secure model for DDS in P2P networks. Section 4.2 presents the summary of [4] in which an access control model for DDS in P2P networks is discussed. Section 4.3 gives a summary of [32] which includes a real example of the use of DDS in a distributed online social networks. Also highlighted are the methods for using the privacy policies of the users to increase data availability for the users in online social networks. Finally, Chapter 5 summarizes the thesis conclusion, and suggests interesting and challenging research directions for future work.

Chapter 2

Storage Data: Distributed Data Structures in Overlay Networks

This chapter addresses the problem of storage of advanced data structures within P2P and overlay networks such as Chord [34]. Such networks only provide the routing and the layer on the top does the replication as well as provide only single data storage which allows access using simple put/get operative commands. In contrast, with the more sophisticated applications such as online social networks that offer more data manipulation and access operatives there will be need for more advanced data structures such as sets, lists and trees to store data. This chapter presents Distributed Data Structure (DDS) as a storage service layer in P2P networks. DDS scheme offers storage services that enable to storage and retrieval of data in a distributed manner. A special replication protocol is implemented to guarantee data persistence and availability. This chapter is broken into three sections. In the first Section 2.1, a summary of [3] is presented that highlights the main methods 2.1 implemented in DDS. The second Section 2.2, is a summary of [6] which introduces a structured cache mechanism with the aim of improving the retrieval time in DDS. Finally, the third Section 2.3, presents a summary of [1], which considers updating mechanisms that ensures the structured cache is kept up-to-date.

2.1 Sets, Lists and Trees: Distributed Data Structures on Distributed Hash Tables

This chapter summarizes the contributions and gives a verbatim copy of our paper [3].

Raed Al-Aaridhi and Kalman Graffi. "Sets, Lists and Trees: Distributed Data Structures on Distributed Hash Tables".

In: *Proceedings of the 35th IEEE International Performance Computing and Communications Conference (IPCCC)*. 2016. Acceptance Rate: 24.7%

In the following we present the summary of our paper [3] in Section 2.1.1. We present the contributions in Section 2.1.2 and personal contributions in Section 2.1.3. The importance and the impact on this thesis is discussed in Section 2.1.4.

2.1.1 Paper Summary

This paper addresses the problem for storing complex data structures in distributed systems such as DHTs and unstructured peer-to-peer overlay networks allow to store, search and retrieve single data elements. This is convenient for simple applications but for advanced applications such as online social networks this becomes not practical because in such applications, for example, each user profile contains a list of friends, box of messages and a set of albums for pictures and each album contains different number of pictures and each picture maybe has many comments and this leads to a big number of data elements for each user. These data elements to be replicated to other nodes in the network via replication services as in PAST [16] in order to keep the data available when the nodes go offline. So, if we want to store all the data elements for the profile in a single node this leads to two issues, first, the replication duration takes a long time, the second issue the single storage node will be overloaded. For that, structured P2P networks are not practical for advanced applications especially the applications which depend on storing complex data structures such as online social networks.

Therefore as a solution for this issue, in this paper, we present our approach which is called distributed data structure (DDS) in P2P networks as shown in Figure 1.2. Our approach is built on the top of DHTs as a storage layer and offers the advanced data structures such as sets, list, and trees. The DDS scheme is composed of three main elements, first compulsory data ID, with these IDs we can store and retrieve the data elements, second, the optional payload for instance picture or specific value. Third, optional pointers which refer to further data elements because DDS scheme is a pointer graph in P2P networks. In our approach we can distribute the load by distributing the storage among the peers in P2P networks, this leads to overcoming the limitation of using structured P2P in the advanced application. The DDS scheme acts as a storage layer located between the storage layer of DHTs and the application layer, this layer is supported from appropriate replication service called simple replication protocol in order to replicate the data items which is stored in DDS scheme among other nodes in DHTs.

The DDS scheme supports four main functions. The first function is called Store DDS. If we want to store any kind of data, we can easily call the DDS layer and this layer will compute the corresponding data structure such as set, tree or list, and compute the ID of each data items in this structure. Then after each data is stored in the single data item and has a specific object ID in DHT, these IDs act as pointers to further data items. This ID considers the entry point for the DDS and can be used later in other operations such as retrieve operation. The second function is Retrieve DDS, we use this function to retrieve the data item stored in P2P networks, the retrieving operation used the object ID as an entry point to the DDS, for instance in case of the tree data structure the entry point is the root of the tree. For the retrieve process in P2P networks, the application layer in DHT can easily call the DDS layer with the object ID. Then after the DDS layer retrieves all the entries one after another recursively, because of DDS scheme are pointer graphs, therefore, we obtained many new pointers and IDs with every entry retrieve. The data items are retrieved via the resolving of the corresponding set of pointers.

The third function is called Modify DDS -Add Entry which represents the add operations, and the fourth function is named Modify DDS - Delete which represents the delete operations. In both operations, the application layer calls the DDS layer to add or delete an entry. The DDS layer computes the corresponding data item that needs to be changed and then in case of the delete operation, the DDS layer requests the storage node in DDS to overwrite the corresponding data item by a special data item marking the deletion of this item, while in case of adding entry, the DDS layer executes the new storage operation with the new data entry for the corresponding object ID of the data entry.

In the evaluation of the DDS scheme, we focus on three matrices in order to evaluate the cost and the quality of our approach. We use an event-based simulator called PeerfactSim.KOM in order to get these matrices. Furthermore, we conducted the investigation in five scenarios and these scenarios represent the main DDS categories such as lists, sets, and trees, in order to get a wide range of ideas and concepts in the behavior of DDS scheme. We evaluate the costs and the performance in small and large scale scenarios, and we use in the experiments churn and without churn, the churn for a realistic environment and without churn to see the fundamental functions of DDS scheme, as well as on deep and broad distributed data structures in a network with 1000 nodes.

The results of the evaluation show that all elements of each DDS are successfully stored and retrieved and the results ensure that our approach comes with low overhead and delay. Furthermore, we remark in the evaluation that the retrieval operations in broad DDS such as a broad tree are very quick, while in deep DDS such as a deep tree the retrieval operations take quite a long time.

2.1.2 Contributions

The contributions are the solving of the problems of storage of complex and big data structures in structured P2P. Therefore, we introduce the DDS scheme which supports advanced data structures such as sets, lists, and trees which support sophisticated applications such as online social networks. In our approach, we investigate the behavior performance of the DDS scheme via implementing and evaluating a different large scale of scenarios such as extreme deep/broad trees as well as a scenario from a social networking which includes storage of a profile, set of the album, and images in an event-based simulator called PeerfactSim.KOM.

2.1.3 Personal Contributions

The contributions of Raed Al-Aaridhi, the author of this thesis, are the implementation, carrying out a set of experiments, drafting the paper, as well as the development of the analysis of the paper and the joint conduction of the experiments. The discussions on the research, the solution and the methodology of testing was done in cooperation of the author of this thesis with Kalman Graffi.

2.1.4 Importance and Impact on Thesis

This paper is an extremely important work and has a high impact on this thesis, and is consider the basic to this thesis for many of reasons since it allows to answer many of the research questions listed in Section 1.1. We consider a new storage scheme and we show with this paper that it is possible to store and retrieve a complex and big data structures in structured P2P networks with high reliability and distributing the load by distributing the storage. Our approach is a storage service layer which utilizes a DHT and is built on the top of it. As we presented in Chapter 1, we investigate the future applications of P2P networks such as online social networks in DHT and overcome the limitation for DHT which only support the single data element storage.

2.2 Distributed Data Structures Improvement for Collective Retrieval Time

This chapter summarizes the contributions and gives a verbatim copy of our paper [6].

Raed 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 (IEEE WPMC '16)*. 2016. Best Paper Award.

In the following Section 2.2.1, we present the summary of our paper [6]. We present the contributions in Section 2.2.2 and personal contributions in Section 2.2.3. The importance and the impact on this thesis is discussed in Section 2.2.4.

2.2.1 Paper Summary

This paper addresses the problem of delay of retrieval operations in distributed data structures. The aim of this paper is to resolve the problem of the time-consuming retrieval of deep data structures [3] which have been shown to take O(log N) hops in the overlay network, as a result of the DHT. Therefore, as a solution to this problem, this paper presents a cache structure, which helps to accelerate the retrieval process by a considerable measure. During evaluation and experiments done on the DDS scheme in [3] considering the deep tree structure as shown in Figure 2.1, it is observed that the collective retrieval duration takes quite long because many pointers need to be resolved iteratively.

This happens because each DDS item only resolves the pointer to the next element, which leads to a chain of individual item retrievals without any parallelization and for each retrieved DDS object a further DHT lookup is required. This then leads to an increase in the average duration time of collective-retrieves, hence demonstrating a performance problem occurring with greater depth in the DDS. Therefore, in order to deal with this issue, in this paper, we present the cache mechanism. This caching mechanism has the overlay keys of deeper DDS objects in advance enabling the client node to finish the complete retrieval of the DDS scheme significantly faster. The client node in our approach can retrieve all DDS scheme objects at once from the DHT without needing to wait to retrieve the objects after each depth-layer. At this point, a Structure Cache was added to the DDS scheme as shown in Figure 2.2, which

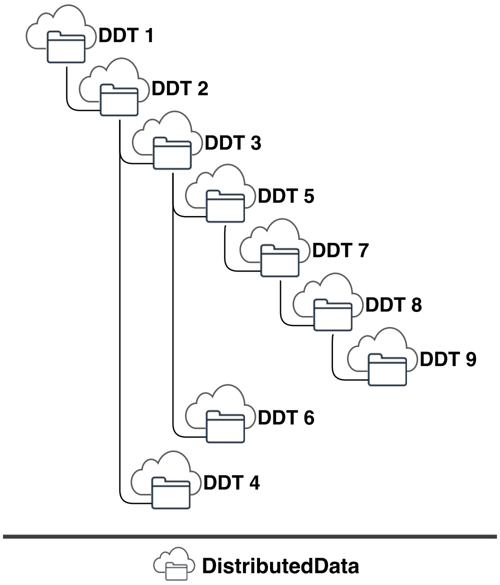


Figure 2.1: Sample Distributed Data Structure: Tree. © [2016] IEEE

was saved as an atomic data along the object of the DDS. In addition, the Structure Cache contains all the overlay keys for deeper DDS objects under the cached root object. So, when we retrieve the root object of Distributed Data Trees (DDT), we get the Structure Cache which is attached to this root object and all deeper DDT objects will start to retrieve simultaneously in parallel. Once the last individual retrieve process is ending, the retrieved overlay key and the DDT pairs are used to rebuild the Structure Cache of the complete DDT, and the complete retrieval is ending

During the investigation of the Structure Cache behavior, one limitation of using the Structure Cache appears about keeping the Cache updated. One would get outdated results if the Cache is not updated quickly enough. Therefore, this issue should be investigated and appropriate solutions should be found such as sending immediately an update command for the Structure Cache after the structure of a DDS has been changed.

In the evaluation, we focus on three main metrics, named storage and retrieval success, retrieval delay, and traffic overhead/number of contacted nodes to measure the quality and cost for using the DDS scheme with and without the Structure Cache. We simulated the DDS with and without churn and measured the retrievability as well as the retrieval delay and traffic overhead via an event-based simulator called PeerfactSim.KOM [17, 21, 25]. In the experiments, we consider the scenario of a deep DDT with 100 entries, in this tree every object except the leaf objects has one single child object. We perform tests both with and without churn to measure the impact of churn. The typical use case for such deep DDT is an article with connected comments on the of a user where every new comment can be attached to the previous last comment, and thus creating an ordered list of comments under the article as in Figure 2.1.

The evaluation results for the DDS scheme show that all operations for DDS scheme are performed correctly with and without Structure Cache both under and without churn in larger

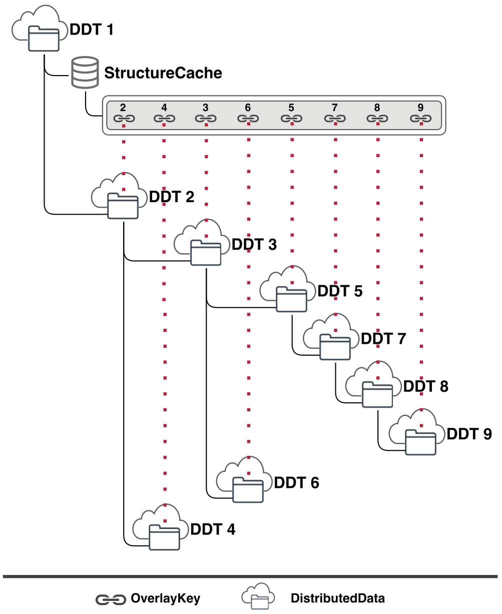


Figure 2.2: DDT with Structure Cache Construction. © [2016] IEEE

setups with 1000 active nodes. Only seldom failing peers or timeouts lead to single operation misses, resulting in a success ratio of the lookup of 0.99% in minimum. Thus the functional requirements of resolving the DDS retrieval are satisfied fully. Furthermore, the retrieval duration with Structure Cache was faster than without the Structure Cache as shown in Figure2.3. During the investigation of the behavior of the Structure Cache, we figure out that the Structure Cache is needed to be kept up-to-date when data is added or changed in the structure.

2.2.2 Contributions

The contributions are the solving of the problems of time-consuming retrieval of deep DDS in structured P2P. Therefore, in this paper, we introduce the concept and benefits of using the Structure Cache for the DDS. The Structure Cache helps us to perform a complete retrieval of the DDS significantly faster via simultaneously in parallel retrievals. Furthermore, we evaluate the DDS scheme with and without churn, as well as we investigate the behavior performance of the Structure Cache via implementing and evaluating extreme deep trees with and without Structure Cache in an event-based simulator called PeerfactSim.KOM.

2.2.3 Personal Contributions

Raed Al-Aaridhi who is author of this thesis, organized and wrote major parts of the paper, conducted the experiments, selected the appropriate evaluation metrics and analyzed the results as well as undertaking of frequent discussions over the solution and the evaluation methodology.

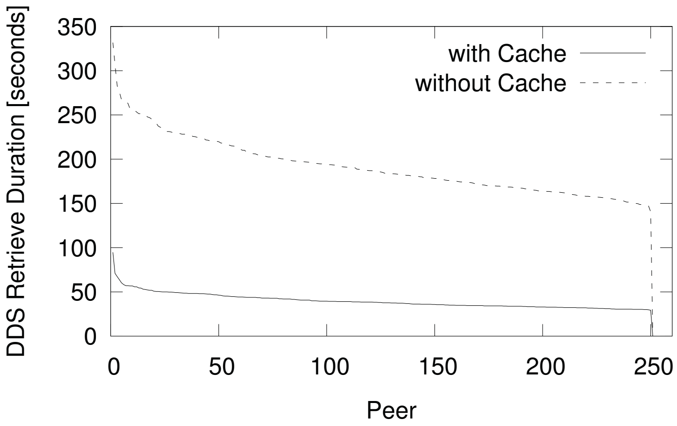


Figure 2.3: DDS Retrieve Duration with Churn. © [2016] IEEE

The implementation and drafting were done by Ahmet Yüksektepe, the paper is presented by Tobias Amft. The discussions on the research, the solution and the methodology of testing was done in cooperation of the author of this thesis with Kalman Graffi.

2.2.4 Importance and Impact on Thesis

In this paper, we present our contributions of Structure Cache which we used in order to accelerate the retrieval operations for deep DDS, which we present in Section 2.1. To achieve this, we attach a Structure Cache to the DDS such as in a deep tree to improve the performance of the data retrieval by maintaining and indexing a data structure at the root of the DDS scheme. We investigated the performance of the Structure Cache applied in the DDS scheme under churn and without churn in a network. Furthermore, we show that the DDS scheme operations with Structure Cache are fully functional and resolve to nearly 100% of the retrieve operations and the retrieval time of DDS scheme with a Structure Cache becomes much faster.

2.3 Optimization of the Structure Cache for Distributed Data Structures in Overlay Networks

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

Raed Al-Aaridhi, Niklas Foerst and Kalman Graffi. "Optimization of the Structure Cache for Distributed Data Structures in Overlay Networks". In: *Proceedings of International Conference on Electrical and Computing Technologies and Applications (IEEE ICECTA '17).* 2017.

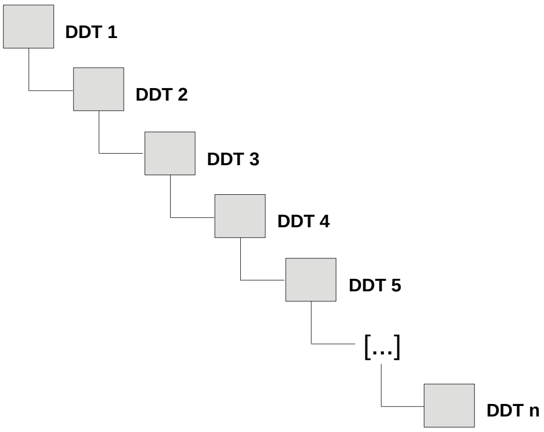


Figure 2.4: Example DDS of a Deep Tree Structure with n data elements. © [2017] IEEE

In the following we present the summary of our paper Optimization the Structure Cache [1] in Section 2.3.1. We present the contributions in Section 2.3.2 and personal contributions in Section 2.3.3. The importance and the impact on this thesis is discussed in Section 2.3.4.

2.3.1 Paper Summary

This paper addresses the problem of keeping a Structure Cache which we introduced in [6] up-to-date. This paper aims to introduce different approaches that ensure the Structure Caches which are used to improve the retrieval of distributed data in P2P systems are kept up-to-date. The DDS scheme in [3] shows in evaluations low overhead and delay. Through the iterative retrieval of the entries, typically multiple nodes provide their single DDS scheme items to the retrieving node.

It is observed during the experiments, in specific considering a deep tree structure as shown in Figure 2.4 that the collective retrieval duration is fairly long due to the use of iterative use pointers to resolve an item and then call the next pointer. In order to retrieve the whole DDS, the collective of data items, each corresponding data item has to be obtained from the network. In the DDS scheme, every peer can only resolve the pointer to the next peer, in case of deep DDS, this leads to a chain of individual item retrievals without any benefits through parallelization. As a solution for this issue, we present in previous work [6] a Structure Cache for the DDS scheme to retrieve the complete DDS scheme much faster. This Structure Cache contains information about the deeper DDS objects under the cache and saves it as an atomic data along the root object of the data structure. The main limitation of the Structure Cache feature is the need to keep it updated when data is added or changed in the structure. Therefore, as a proposed solution to this issue, we present in this paper two approaches for updating Structure Caches in Distributed Data Trees (DDTs).

The first approach is named the Periodic Top-Down Update. In this approach, we should check periodically if the Structure Cache is up-to-date, to realize that, we retrieve the whole structure, starting at the data element at the node, which is responsible for the Structure Cache that we

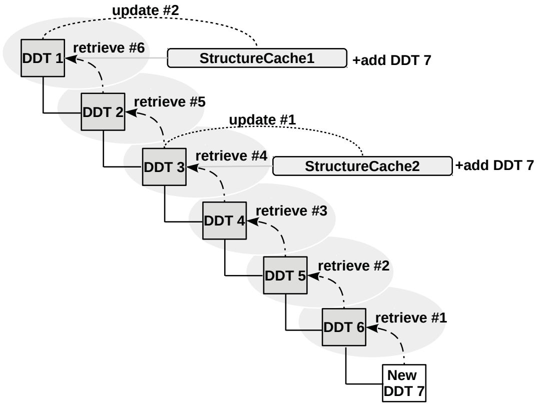


Figure 2.5: Example for the Instant Bottom-up Update mechanism. © [2017] IEEE

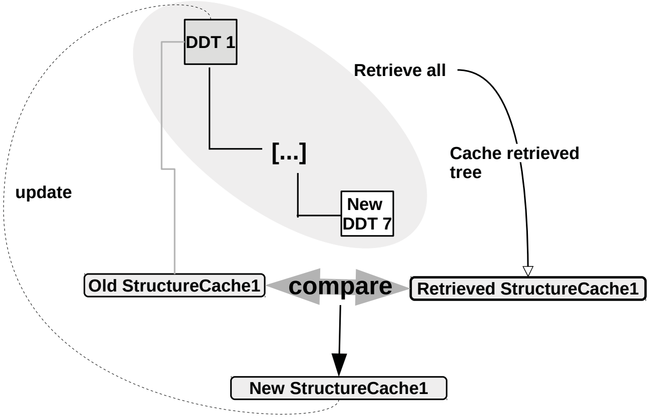


Figure 2.6: Example for the Periodic Top-down Update mechanism. © [2017] IEEE

want to check. This can be done either via the node that is storing the Structure Cache, in that case, there are fewer retrievals or done by any other node such as the node that is the owner of the stored data. We decided in our approach that the node which initially stores the DDT is responsible for the update as shown in Figure 2.6. This updating approach is suitable to use for the Structure Cache, in case the Structure Cache does not get any information from the nodes if the data is changed or not.

The second approach for the update mechanism is the Instant Bottom-Up Update. In this approach, the update is done through backtracking the data structure from the newly added element to the root. To do that, every data element has to save its own predecessor/parent as well as its children. In this approach, the update mechanism starts when the data element is added or changed in the structure. With this update mechanism, we retrieve only the predecessor of the changed data element and check if the element is in charge of a Structure Cache. In case no cache is available, we retrieve the next predecessor and so on. This approach uses additional information about the predecessors in a DDT to backtrack the structure to its root as shown in Figure2.5

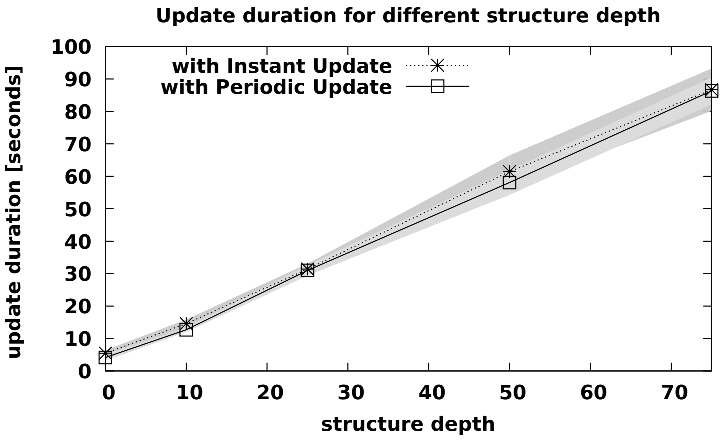


Figure 2.7: DDS Retrieve Duration with Churn. © [2017] IEEE

In the evaluation, we focus on Retrieval delay in order to evaluate the time needed to deploy and retrieve DDSs which uses the Structure Cache in relation to their size by using both updating mechanisms with different deep trees. In the evaluation process, two simulation series is performed, one series each for the Periodic Bottom-Up Update method and the Instant Top-Down Update. In the experiments, for every single run, the depth of the DDT is changed, changing it in different ranges, in order to compare the measurements with different structure sizes. A node starts by adding a data element at the bottom of the tree structure.

When the data element is added, one of the two update mechanisms was then started. To avoid measurement errors due to randomness, different seeds that PeerfactSim.KOM provides were used and the experiments repeated. The results of the evaluations show that both update duration times are nearly identical as shown in Figure 2.7. Both grow linearly because both mechanisms behave in a similar fashion. The whole DDT retrieves element by element in both update approaches. In the case of Periodic Top-Down Update, the update process starts at the root to obtain the information about the whole DDT. On the other side, in the Instant Bottom-Up Update, the update process starts via retrieves the element from the bottom to find all possible Structure Caches.

During the investigate of both approaches, it appears that one of the advantages of using the Instant Bottom-Up Update to update the Structure Cache is, that it only has to update once and in this progress, it also updates all caches along its way. This comes at the cost of high update times for larger structures. Nevertheless, the advantages could overcome this drawback in certain applications. While the update time for the Periodic Top-Down Update grows significantly less, but even with this approach, we should keep in mind that it is a periodic process, and this leads to the possibility of waiting for a certain period until the update occurs. In addition, adjust the update interval depends highly on the needs of the application.

2.3.2 Contributions

The contributions are the solving of the problems of the Structure Cache, in particular when the Structure Cache becomes outdated. Therefore, in this paper, we introduce the concept and benefits of using the update methods for the Structure Cache in DDS scheme. As a next step, we show through simulation the performance characteristics of updating methods. Finally, we evaluate and compare different kinds of updating methods.

2.3.3 Personal Contributions

The contributions of Raed Al-Aaridhi are the provision of the idea, methodology and design of the paper, the selection of metrics as well as writing the major part of the paper. The implementation as well drafting were done by Nikolas Foerst, the part of discussion related to the solution and the methodology was done in cooperation of the author of this thesis with Kalman Graffi.

2.3.4 Importance and Impact on Thesis

This paper is an answer to the research question on how to improve of the performance of data retrieval in a deep DDS scheme. In this chapter, we present different approaches in order to overcome the limitation of keep the Structure Cache up-to-data. We identify two approaches in order to keep the structure cache up-to-date with this work, and we elaborate the advantages and drawbacks of both update mechanisms. The new application can choice any approach, only depending on how application uses the Structure Cache and what the main purpose of the application is.

Chapter 3

Services for Distributed Data Structures in P2P Networks

P2P overlay networks consist of many logical layers which use and get benefits from the functions that are offered in the under-layers and at the same time generate services in order to add new functionality and improve the quality of the P2P applications.

The focus of this chapter is the services and functionality of the distributed data structures which are stored in P2P networks. The previous Chapter 2 addresses the problem of storage of advanced data structures within P2P and overlay networks because of the limitations inherent, specifically, single data element storage, such as in Chord [34]. As a solution, the Distributed Data Structures scheme (DDS) [3], [6] and [1] is introduced which can be implemented in the P2P networks and offers a storage layer for future applications that require the capability of storing advance data structures such as sets, lists and trees with guaranteed data availability through an appropriate replication mechanism.

This chapter addresses the challenges of adding services such as search mechanisms of metadata as well as adding computational data elements to DDS scheme in order to improve the quality and adding new functionality to the DDS scheme. Metadata can be defined as data or information about a file or document for example the name of an author or the title of a book and the functionality of the DDS will essentially be incomplete if an efficient mechanism for searching metadata is not considered. Therefore, we present in Section 3.1 and [7] several search mechanisms for metadata that are integrated with DDS in P2P networks.

Computational elements are important in sophisticated applications and in order to add new functionality to the DDS and extend the use cases in distributed applications in a heterogeneous environment and ensure that the values are up-to-date, with correct function results, we introduce an extension for the DDS called the Computational Data Element (CDE), which interprets and computes the DDS payload. This is discussed in Section 3.2 and [2].

3.1 Search Algorithms for Distributed Data Structures in P2P Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [7].

Raed Al-Aaridhi, Iakov Dlikman, Newton Masinde and Kalman Graffi. "Search Algorithms for Distributed Data Structures in P2P Networks". In: *Proceedings of the International Symposium on Networks, Computers and Communications (IEEE ISNCC)*. 2018.

In the following we present the summary of our paper on the search algorithms [7] in Section 3.1.1. The contributions to the research are enumerated in Section 3.1.2 and personal contributions are presented in Section 3.1.3. The importance and the impact on the thesis is discussed in Section 3.1.4.

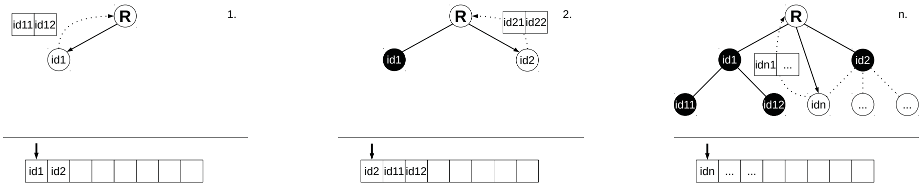
3.1.1 Paper Summary

In this paper, we introduce algorithms that support metadata search in structured P2P overlay networks such as Chord [34]. In previous work such as in [3], [6] and [1], a DDS scheme based on DHTs is proposed, which consist of linked data items. This DDS is basically a pointer graph in the DHT in which each linked data object contained ID information, pointers, and a payload. Further, the DDS supported key functions such as Store DDS, Retrieve DDS, Modify DDS (that is, Add Entry and Delete Entry). The proposed DDS scheme's evaluations had low overhead as well as low delay. It is noteworthy, that the functionality of this DDS will essentially be incomplete if an efficient mechanism for searching metadata is not considered. Therefore this work extends the DDS scheme proposed in [3] by supporting metadata search algorithms to add the search functionalities. We propose four solutions to perform the metadata search namely, LocalJoin, ParallelJoin, NetworkJoin and BloomJoin and the root node considers as an entry point for the search in all algorithms. Below, we discuss briefly each approach.

The first approach is LocalJoin, in this algorithm the join operation takes place at the local node. The search starts in this approach from the root node, the root node has a list and it is copying of all the ids for its children to this list one after the other as shown in Figure 3.1. The importance of the pointer is that it refers to the search steps between the entries. The algorithm is ending either when matching nodes regarding the search criteria are found or when there are no IDs available in the list because this means there is no node left that has not been searched.

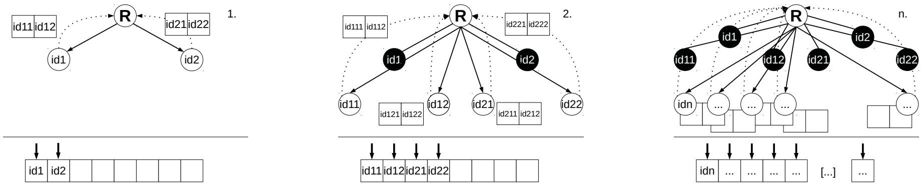
The second approach is Parallel LocalJoin, in this algorithm, we develop the concept of Simple LocalJoin approach by using the principle of parallelism in the search. So the root node contacts its children nodes in parallel and retrieved sequentially as shown in Figure 3.2.

The third approach is Asynchronous NetworkJoin, in this algorithm, the root node gives control to its child nodes to perform the search. The search starts from the root node via sending asynchronously a query message to each of its children as shown in Figure 3.3, and when any



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

Figure 3.1: Simple LocalJoin. © [2018] IEEE



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

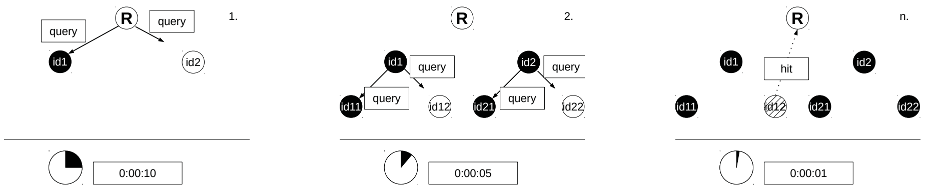
Figure 3.2: Parallel LocalJoin. © [2018] IEEE

node matches the search criteria, it will then sends a Hit message to the root node, and the root node will stop the timer and the algorithm changes its status from in progress to finished successful when it receives such a Hit message. In this approach, the timer determines how long the root node will wait until the search is completed.

The fourth and last method is BloomJoin, in this algorithm in each step, more nodes are involved in searching a rising amount of ids and the root node accesses the search results at each step through employing remote method invocation (RMI) [18, 29]. In every step, the nodes gain access to further nodes as shown in Figure 3.4. The algorithm stops either when matching nodes regarding the search criteria are found or no new nodes are left.

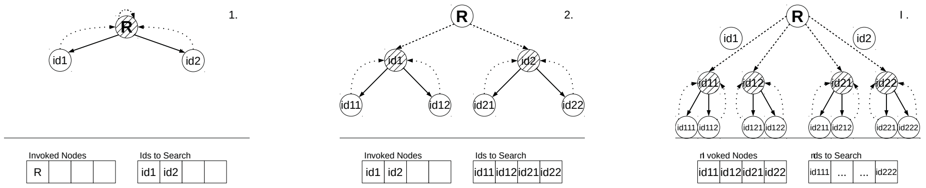
In the evaluation, we focus on six metrics in order to evaluate the quality and costs of the search mechanisms in DDS scheme, we focus on the amount of traffic generated for each search process by bandwidth, the operation duration taken for each algorithm to search through the whole DDS, hop count, number of messages, number of contacted nodes and measures of the reliability of the search algorithms. In addition, we evaluate and implement three Distributed Data Tress (DDT) scenarios In order to obtain a broader set of ideas in the behavior of search algorithms, and each of three DDT types with a group of 1000 participants, the scenarios such as the binary tree as in Figure 3.5a, deep tree as in Figure 3.5b , and customized broad tree as in Figure 3.5c. We implement two kinds of search techniques, Exhaustive Search, and First Match Search. In the Exhaustive Search, the algorithm proceeds through the entire DDS ensuring that it exhaustively searches every node. The First Match Search, on the other hand, terminates the search process as soon as a suitable match has occurred.

The evaluation results of the Exhaustive Search, as well as the First Match Search for all algorithms, show that the LocalJoin algorithm is the best with regards to all the six evaluation metrics even under the churn. Next is followed by the BloomJoin algorithm which had good results in term of duration when considering the First Match Search experiments for the binary



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

Figure 3.3: Asynchronous NetworkJoin. © [2018] IEEE



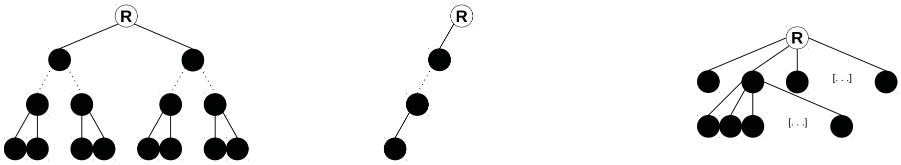
(a) Step One (b) Step Two (c) Step N

Figure 3.4: BloomJoin. © [2018] IEEE

and deep tree, as well as appears to be good performers in terms of traffic. In addition, the results show that the NetworkJoin algorithm has the worst performance compared with the other algorithms when evaluating all metrics together. We observed during the experiments that all algorithms except the Localjoin have a weakness on a broad tree topology and they fail to terminate the search, this issue needs more investigation in future works. Furthermore, the performance of NetworkJoin can be improved via determining the number of forwarded messages by checking the recipient of the query messages, and in order to make more specific statements about the termination result of the algorithm, we can also adjust the timer of the timer mechanism. In the end, we believe that our search algorithms for metadata will benefit the creation of further P2P-based applications.

3.1.2 Contributions

The contributions are the solving of the problems in Challenge 3 which is adding new functionality of the DDS scheme by introducing the concept and benefits of using search mechanisms for searching metadata in the DDS scheme over dynamic overlay networks. Therefore, in this paper, we propose different search methods, specifically, LocalJoin, Parallel LocalJoin, NetworkJoin and BloomJoin, which are implemented in such networks so as to support metadata searches. Furthermore, we show through simulation the performance characteristics of search methods.



(a) Binary Tree (b) Deep Tree (c) Customized Broad Tree.

Figure 3.5: Tree Structure used in Experiments. © [2018] IEEE

3.1.3 Personal Contributions

Raed Al-Aaridhi who is author of this thesis, organized and wrote major parts of the paper, conducted the experiments, selected the appropriate evaluation metrics and analyzed the results as well as undertook frequent discussions over the solution and the evaluation methodology. The implementation and drafting were done by Iakov Dlikman, Newton Masinde drafted and reviewed the paper. The discussions on the research, the solution and the methodology of testing was done by the author of this thesis in cooperation with Kalman Graffi.

3.1.4 Importance and Impact on Thesis

This work answers the research question on how the search mechanisms in the DDS storage scheme supports metadata searches and which searching methods are appropriate for use with metadata-based search. We show with the search mechanisms the possible ways to realize searching with DDS in an overlay network such as in Chord [34]. In this way, overlays and applications on top of the DHT can process this information about searching to provide further services.

3.2 Computational Elements For Distributed Data structures On Overlay Networks

This chapter summarizes the contributions and gives a verbatim copy of our paper [2].

Raed Al-Aaridhi, Felix Bandosz, and Kalman Graffi. "Computational Elements For Distributed Data structures On Overlay Networks". In: *In Proceedings of Academics World International Conference (AWIC)*. 2017.

In the following we present the summary of our paper Computational Elements [2] in Section 3.2.1. We present the contributions in Section 3.2.2 and personal contributions in Section 3.2.3. The importance and the impact on this thesis is discussed in Section 3.2.4.

3.2.1 Paper Summary

This paper focuses on the problem of how to compute the payload stored in the P2P networks with correct functions as in the scenario of Bob's profile as we explained in Section 1.2 and how we compute the payload as in Figure 1.3. The aims of this paper are to adding new functionality to the DDS scheme [3], [6] and [1], also to integrate to the DDS scheme with the computational elements, which allows the dynamic updating of the values in P2P networks. We presented in the previous Chapter 2 DDS scheme which offers a storage layer on P2P networks. The DDS scheme is composed of IDs, values, and pointers to further nodes as pointer graphs in the DHT.

Nowadays, many advanced applications such as complex event processing need to resolve the functions correctly because in such applications we have inputs, outputs and some functions which perform processing information in the network that we actually call computational approach. It has nodes with different functions, some nodes only generate the data such as sensors, while the other nodes perform data analysis on the generated data.

In order to resolve the functions in the distributed systems such as P2P networks as well as extend the use cases of the DDS scheme, in this paper, we introduce the Computational Data Elements (CDE) as an extension to DDS scheme. To achieve that, we implement the backtrack to DDS scheme as shown in Figure 1.3 Part C in order to be able to exchange the values between the nodes in the DDS scheme. For example when the leaf child is changing the value then the parents nodes which are using this leaf child as input should change its value to be updated again. In our approach the elements are combined with the links to other functional elements and the graph of functional elements is created via these links which lead to a linked data graph.

The CDE is consisting of an identifier, a function of time, or a function over further function. The values in CDE are changing sometimes through a function of time such as A = f(t), in this case, is over time changing or is a function requiring further function such as A = B + 1, in this case, the value of A will change when the value of B is changing. In addition, in our approach, we use the concept of subscriptions and we present two kinds of subscriptions, namely manual and automatic in order to resolve the CDEs in the networks because all the values, as well as the results of the functions required, must be retrieved by the CDE function to solve them. To achieve this retrieval, we present the idea of subscribing a value which means pushing information to nodes that are interested in a specific value, so by subscribing to the corresponding node will get always the update value from the responsible node immediately. In CDEs, in most cases we have one function depends on the output for other functions and that depends on the output of other functions so we have a chain of functions such as A = Value, B = A + 1, C = B + 1, D = C + 1, E = D + 1, F = E + 1, so when the value of A changes over the time, the values of B, C, D, E, and F will change. To achieve that, we implement four update protocols for the CDEs such as pull/push the value periodically, reactive and proactive. The protocols of pull/push the value periodically are meant that for example, every X second the storing node pushes or pulls the value. The proactive protocol means when the value is changed, the storing node pushes the value. While the reactive protocol means when the value is needed the pull is happening.

In the evaluation, we focus on three main metrics in order to evaluate the quality and costs of the computational methods. We focus on the correctness ratio in order to know the correctness of the distributed functions, the error happens in CDEs when the input values are changed and the function itself does not recognize the change. The second metric is the sum of messages in order to count the messages sent between the nodes in every distributed method, while the third metric is the sum of initiated lookups for each method.

The evaluation results for CDE with the DDS scheme show that the proactive method is the best in term of correctness and it has the highest correctness ratio around 95%. In this method when the values are changed, the updating is happening. On the other side, the results show that the reactive method is the worst, in this method the value is pulled only when it is needed. The pull periodically approach consumes the most messages because in this approach the messages are sent based on a timer, for example, every second. It is followed by push periodically approach, which consumes less than pull periodically approach, because of the pull periodically approach must request for each value. The results show that the proactive and reactive methods are more efficient because they consume fewer messages. In term of the lookups, the reactive approach is the best and it has the fewer lookups while the pull periodically approach has the most lookups. This work ensures that the proposed distributed methods which are integrated with the DDS scheme can benefit the creation of further P2P-based applications.

3.2.2 Contributions

The contributions of this paper are the solving of the problems in Challenge 4 which is regarding the implementing of computing methods in the DDS. Therefore, in this paper, we identified the concept and benefits of using computational elements in DDS within P2P networks as they add new functionality to the DDS scheme [3] and allow for ease in extending the use cases of DDS in distributed applications. We propose an extension for DDS called CDE which interprets and computes the DDS payload with correct functions by implementing four new distribution methods called: “push without timer“, “push with timer“, “pull without timer“ and “pull with timer“. We show through simulation the performance characteristics and compared different kinds of computational methods.

3.2.3 Personal Contributions

Raed Al-Aaridhi, who is the author of this thesis, organized and wrote major parts of the paper, he identified the computational approaches, designed the experimental work, selected the metrics for the evaluations and undertook the dissection of the implementations. Felix Bandosz implemented the work and participated in the drafting. The discussions on the research, the solution and the methodology of testing were done in cooperation by the author of this thesis with Kalman Graffi. Newton Masinde supported the work through final reviewing and improving of the language.

3.2.4 Importance and Impact on Thesis

This work answers the research question on how we can resolve the functions as well as implement the distributed methods in DDS scheme [3], since in most cases functions need value from other nodes, this leads to adding new functionality as well as extends the use cases for DDS scheme in advanced distributed applications. We show with the computational elements the possible ways to realize how we merge computational methods with DDS in P2P and overlay networks such as in Chord [34].

Chapter 4

Security, Privacy and Access Control for Distributed Data Structures in P2P Networks

In the recent times, P2P networks are used in sophisticated applications such as in the field of high-performance computing and in online social networks especially because P2P networks overcome some major limitations of client-server networks such the scalability, single point of service and central authority. The P2P networks provide open and public networks which depend on the participation and collaborations between users/peer. This means that any peer can join and participate, even the malicious peer, which can potentially be harmful for other peers in the network, for example, by blocking of essential services or modifying the data.

In this chapter, focus is on the third part of the scope of this thesis: security and access control for distributed data structures (DDSs) which are stored in P2P networks. Furthermore, an application for utilizing content allocation strategy that supports content availability in a Distributed Online Social Network (DOSN) is proposed and implemented in Chapter 4.3. This application ensures that there is a level of privacy enforced during the replication of data. We integrate our security mechanism for distributed data graphs, which guarantees safety of the data in P2P networks via a secure model, which is introduced in Chapter 4.1. The secure model is extended by introducing access control mechanisms in the DDS scheme which is discussed in Chapter 4.2. In Chapter 4.3, a data allocation approach for DOSNs is presented, which uses the privacy policies of the participants to increase the availability of the participants' contents without diverging from their privacy preferences is presented. This work builds on the DDS scheme [3], [6] and [1] which is implemented within P2P networks.

4.1 Secure Model for Distributed Data Structures on Distributed Hash Table

This chapter summarizes the contributions and gives a verbatim copy of our paper [5].

Raed Al-Aaridhi, Ahmed Yüksektepe and Kalman Graffi. "Secure Model for Distributed Data Structures on Distributed Hash Table". In: *Proceedings of the IEEE Jordan Conference* *on Applied Electrical Engineering and Computing Technologies (AEECT)*. 2017.

In the following we present the summary of our paper [5] in Section 4.1.1. The contributions to this work are discussed in Section 4.1.2 with the personal contributions in Section 4.1.3. The importance and the impact on this thesis is brought out in Section 4.1.4.

4.1.1 Paper Summary

This paper addresses the problems of security of DDS in P2P networks. P2P networks are open and public networks, exposing the networks to malicious peers who can exploit and damage the networks in different ways such as data misuse and spy on other peers, spreading of viruses and so on [36]. Therefore, peers cannot trust each other and the security of data is an essential topic in P2P networks because proper security mechanisms ensure that it is possible to store private and sensitive data at unknown network peers since the stored data cannot be easily read or manipulated. The paper presents a secure model for DDS in P2P systems. In Chapter 2, previous works [3] are discussed, introducing a DDS scheme on DHT, which consists of linked data items which contain (ID, optional pointers/payload) within P2P networks. The data storage mechanism in DDS that is introduced among other things requires appropriate security mechanisms to ensure secure data storage. In this paper, we present and evaluate a concept of a secure model which works completely without trusted nodes for such DDS in P2P networks.

In the beginning, we identify the expected functional requirements and the assumptions for the secure model. The requirements such as security of data, no trusted third parties and continuous security. We assume that each member is probably a malicious node. In addition, we assume that the users have the public keys of their friends and other known users as well as they are able to demand any ownership over a distributed data in the P2P network. Using these requirements, we define what features the secure model should offer to the DDS scheme.

In our paper, we use many security mechanisms such as the hybrid cryptography, granular encryption and granular signature (GEGS), and the security service. We use the hybrid cryptography's great features which is combining the advantages for asymmetric and symmetric cryptography. So, first, we encrypt the data with the symmetric encryption key and after that, we encrypt this key with public keys for users that are accessible, in this way, we are able to give the access right for specific users, then after, we can keep the encrypted data together with the asymmetric encryption of the encryption key per user. In our DDS scheme, the creator of the distributed data is the only one who who can generate the encryption key. In our approach, we termed the data is secured if it has been encrypted and the encrypted form is signed by an accredited user. On the other side, the decrypting as well as the verification of the signed data is termed the Desecuring.

In each DDS, we have three main items such as ID object, payload, and pointers to other data items, and the security mechanism will be not efficient if we protect only the value, therefore, we need to protect all items in DDS. To achieve that, in this paper, we introduce the GEGS for protection of all the attributes of DDS via the deal with the DDS as a set of various

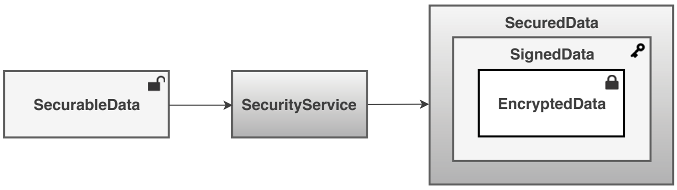


Figure 4.1: The concept of Secured Data for Distributed Data Structures (DDS). © [2017] IEEE

secured data items, and this leads to prevent the malicious users from getting any information concerning the DDS scheme.

In the implementation, the security service is in charge of the conversion of unprotected data which is termed in our approach as securableData into a secured data as illustrated in Figure 4.1. With security service, we achieve the required continuous security, because we use the last version of asymmetric and symmetric for encryption/decryption operations as well as the last version of the cryptographic hash function, so for example when RSA-2048bit becomes not trusted in the future, at that time we can change it easily and use the updated release in the security service.

In the evaluation, we focus on the performance costs and to measure the overhead of time and space caused by the DDS secure model. Therefore, we conduct four experiments in our evaluation of the security model for the DDS scheme. In the first experiment, we focus on the evaluation of the time needed for the validation and decryption operations in the secure model. In the second experiment, we focus on the overhead caused by the security techniques which are used in the security model, to achieve that, we compared the retrieval time for the DDS without security mechanisms against retrieval time for the DDS with the security mechanisms (including time needed for the validation and decryption operations). In the third experiment, we focus on measuring the storage space which is needed to store the DDS as well as its secure model. In the last experiment, we focus on measuring the impact of adding a new privileged user to the whole secured storage space.

The evaluation results show that the security mechanisms which are used in the secure model for the DDS scheme add an unavoidable overhead to the time and storage space, and these costs are still acceptable compared with the set of features offered by the secure model. We believe that the security mechanisms which are introduced in this paper and are integrated with the DDS scheme can benefit the creation of further P2P-based applications.

4.1.2 Contributions

The contributions in this paper are the solving of the problems of the security for the DDS scheme. First, we present the flexible security architecture for DDS scheme based on strictly defined security requirements. Because the DDSs which we present in the first chapter of thesis are without any security mechanism and unprotected, which mean that any peer/user in the network can read, delete or modify the data storage. Therefore, in this paper, we introduce a secure model which allows to store distributed data in P2P networks securely. As a next step, we propose an access control mechanism working completely without trusted nodes. Furthermore, we show through simulation the performance of security architecture. All of these contributions open the door for more P2P applications such as online social networks.

4.1.3 Personal Contributions

Raed Al-Aaridhi who is author of this thesis, organized and wrote major parts of the paper, conducted the experiments, selected the appropriate evaluation metrics and analyzed the results as well as undertook frequent discussions over the shaping solution and planning the evaluation methodology. The implementation and drafting were done by Ahmet Yüksektepe. The discussions on the research, the solution and the methodology of testing was done in cooperation of the author of this thesis with Kalman Graffi.

4.1.4 Importance and Impact on Thesis

This work answers the research question on how to store the distributed data in P2P networks securely. In this work, the DDS scheme proposed [3], [6] and [1] is extended by a secure model based on strictly defined security requirements supporting distributed data security. We show with the security model the possible ways to realizing storage of distributed data objects and ensuring their authenticity, integrity and confidentiality in P2P and overlay networks such as in Chord. In this way, P2P network applications on top of the DHT can use this mechanism about securing data to provide further services.

4.2 Access Control for Secure Distributed Data Structures in Distributed Hash Tables

This chapter summarizes the contributions and gives a verbatim copy of our paper [4].

Raed Al-Aaridhi, Ahmet Yüksektepe and Kalman Graffi. “Access Control for Secure Distributed Data Structures in Distributed Hash Tables”. In: *Proceedings of International Symposium on Local and Metropolitan Area Networks (IEEE LANMAN)*. 2017.

In the following we present the summary of our paper [4] in Section 4.2.1. The contributions are given in Section 4.2.2 and the personal contributions are highlighted in Section 4.2.3. Finally, we discuss the importance and the impact on this thesis is discussed in Section 4.2.4.

4.2.1 Paper Summary

This paper addresses the problems of access control of the DDS in P2P networks. In our paper [5], we introduce a flexible secure model relying on precisely defined security requirements that support distributed data security and ensuring that the data which is stored in DDS scheme is kept secure. In [5] we use many security mechanisms such as hybrid cryptography and GEGS strategy in order to ensure the security of the distributed data objects in P2P networks. The secure model in [5] lacks to address the problems of access control for the data stored in the DDS scheme.

Therefore, in this paper, we extend the secure model in [5] by introducing the access control mechanism for the DDS in the secure model. To achieve that we introduce an access control model named simple security context model. With this model, we are able to secure all attributes of a DDS scheme through the hybrid cryptography. The security context is composed of three main parts as shown in Figure 4.2. The first part is named access control map, in this map, we save all access control information and it is executing the concept of an access control list [30]. This leads to store the public key for favored users in a list called access control entries and the first entry in this map for the creator of the DDS and he has the full access right as well as the control over the DDS. This access control entry has very important information regarding the access right, for instance, give a specific user the access right to the distributed data which is related to the security context as shown in Figure 4.2. In addition, it contains the encryption key for DDS and this key appears only for a special user such as the creator of DDS.

The second part of security context as shown in Figure 4.2 is the secured data items, which is a map for storing the secured form of a DDS according to the name and attribute for the DDS. While the third and last part of security context is named the configuration module, this module is responsible for the management of the encrypted keys, which enables us to choose the encryption key that is needed to perform the new adjustments for the DDS.

Based on the access control mechanism, we are able to solve many access issues such as to protect the identity of privileged users as well as the access revocation issue. To explain the access revocation problem, let us take this scenario, where the user Alex is the creator of the DDS and he gave his friends the access right to his DDS via the access control mechanism and one of his friends named Felix, later on, Alex wants to prevent Felix to access to the DDS but Felix still has the encryption key for DDS and this key enable him to read the new DDS without permission, and it is unpractical to re-encrypt all DDS with new key. Therefore, in order to prevent Felix to access to DDS, we use an efficient method named lazy revocation [28], by this method we need only to change the encryption key based on the configuration module in the security context, and with the new modifications on the DDS, Alex will use the new encryption key, and in this way Alex does not need to re-encrypt all DDS and at the same time Felix will block the access to the new modifications on the DDS because he has not the encryption key.

As a conclusion, in this paper we introduce an access control mechanism for the DDS scheme, this mechanism helps us to solve some access problems. On the other side, our mechanism needs more investigation in order to see how it is secure, in particular the access control map needs more investigation in order to see how it can be secured against hackers and ensure for

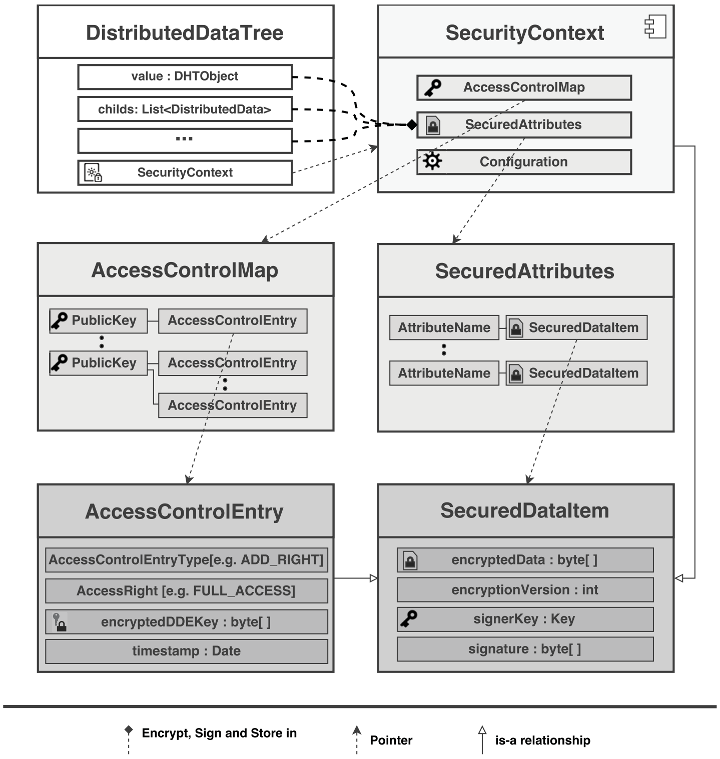


Figure 4.2: A Structure of the Security Context for Distributed Data Structures (DDS). © [2017] IEEE

instance that the entries in the map cannot be changed.

4.2.2 Contributions

The contributions are the solving of the problems in Challenge 5 which are regarding the access control mechanism for the DDS scheme. In our paper [5] we introduce a secure model rely on precisely defined security requirements ensuring that the data which is stored in DDS scheme kept secure. The secure model in [5] lacks to address the issues of access control. Therefore we introduce in this paper an access control mechanism working completely without trusted nodes. For this purpose, different aspects of access control are discussed and initial assumptions concerning the security of the DDS project and its data environment are declared. Many access problems are carefully analyzed such as the revocation problem of shared access to secured distributed system.

4.2.3 Personal Contributions

The contributions of Raed Al-Aaridhi who is author of this thesis are organizing and writing major parts of the paper, conducting the experiments, selecting the appropriate evaluation metrics and analyzing the results. The implementation and drafting were done by Ahmet Yüksektepe. The discussions on the research, the solution and the methodology of testing is done in cooperation of the author of this thesis with Kalman Graffi.

4.2.4 Importance and Impact on Thesis

This work answers the research question on how to extend the functionality of the security architecture for the DDS scheme proposed in the previous Chapter 2 via extending the security model proposed in [5] by a supporting access control model. We show with the access control mechanisms the possible ways to realize how we solve the problems for access rights to distributed data in P2P networks.

4.3 Privacy Preserving Data Allocation in Decentralized Online Social Networks

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

Andrea De Salve, Paolo Mori, Laura Ricci, Raed Al-Aaridhi, and Kalman Graffi. "Privacy preserving Data Allocation in Decentralized Online Social Networks". In: *Proceedings of International Conference on Distributed Applications and Interoperable Systems (IFIP)*. 2016.

In the following we present the summary of our paper privacy data allocation [32] in Section 4.3.1, we present the contributions in Section 4.3.2 as well as the personal contributions in Section 4.3.3. We introduce the importance and the impact on this thesis is discussed in Section 4.3.4.

4.3.1 Paper Summary

In this paper, we propose a new method for the replication of the data in advanced applications such as distributed online social networks (DOSN). In our approach, we utilize the privacy policies of participants in the network in order to increase the availability of participant data. One of the features of the DOSN is that the participants' profiles' data is under control of the peers that build up the DOSN. This leads to new security problems regarding the privacy and availability of participant data. In DOSN, the privacy of published content in the participant profile is very important and should be protected because it contains private information and sensitive data.

This paper introduces a new method of maintaining the participant privacy as well as increasing the availability of participant' data without needing to encrypt the data. To achieve that, our method is composed of modeling the data belonging to the profile of participant X utilizing a pyramidal tree from the DDS scheme which we present in [3], and follow an appropriate strategy to assign the nodes of the tree over the peers of friends of X, and these peers should be available online.

Our approach is to assign a copy of each data D of the tree to another peer who is presently online, and whose participant Y is allowed to access D depending on $X'S$ privacy policy. Therefore, with our approach we avoid the encryption which is necessary to protect the confidentiality of $X'S$ data when stored on $Y'S$ peer, because Y is a friend of X and X based on his privacy policy gave Y the right access to his data.

We presume in our approach, a one-to-one mapping between participants and their peers and we utilize mutually the terms peer or participant to point them out. We propose an approach which is intended to automatically enforce the access control on the basis of the privacy policies defined by participants on their data, by taking into account that resources are organized as a pyramid tree.

Each participant X is linked to its descriptor X, this descriptor has the information regarding the IP address of the corresponding peer, state of the peer either it is online or offline, and it is stored on a DHT and probably retrieved by using the identifier of X. In addition, we assume that all the peers which are able to access the profile of X when X is offline should know the descriptor X. The functions of DHT in our approach are, keep track of the data replicas, the boot of peer-to-peer, and support the search for new friends.

In order to examine our approach, we identified five reference policies which depend on the friendship relationship between the participants, for instance in the first policy, we identified that only participants who have a friendly relationship with participant X can read D. While in the second policy, we identified that participants are able to read D if they have a friendly relationship with participant X as well as have N common friends with direct friends of participant X.

In the evaluation, we focus on the availability of the data after using our approach. To do that, we determine different number of replicas for each data. Furthermore, we examine a load of replicas as well as the time duration for each operation. In the simulation, we initiated a network of 3000 peers, and these peers resemble a relationship such as in a social circle. The evaluation results show that our approach increases the availability of the profiles' data and contents in the DOSN. One limitation appears of our approach that is depending on the availability of trusted replicas, without trusted replicas we will not be able to copy the contents to other peers.

4.3.2 Contributions

The contributions are the solving of the problems in Challenge 6 which is regarding the replicating and increasing the availability of data in DOSN. Our approach depends on the privacy policies of the participants to replicate the contents of participants to other participants. In addition, we propose several algorithms and reference policies in order to realize the contents allocation strategies. Finally, we show the effectiveness of our approach based on the experimental results simulations obtained from on traces is taken from a real social network.

4.3.3 Personal Contributions

The contributions of Raed Al-Aaridhi, the author of this thesis are, the development of a part of this work, in particular the in Distributed Data Tree which used in the implementation as well as frequent discussion with Andrea De Salve who is the main author of this paper. The major work on the paper was performed by Andrea De Salve. He initiated the project and implement the content's allocation strategies which include several algorithms and reference policies, and the paper is with cooperation with Paolo Mori. The discussions on the research, the solution and the methodology of testing was done in cooperation of Andrea De Salve with Kalman Graffi and Laura Ricci.

4.3.4 Importance and Impact on Thesis

This paper is important work and has a high impact on this thesis because he proves the ability to use our DDS scheme in sophisticated applications. This paper answers the research question on how to use DDS scheme in a real sophisticated application such as DOSN. In this paper a Distributed Data Tree is implemented as well as a strategy for a DOSN that exploits the privacy policies of the users to increase the availability of the participant's contents without diverging from their privacy preferences.

Chapter 5

Conclusion and Future Work

P2P networks can be used in many sophistical applications nowadays as they embody the concept of decentralization, which overcomes many disadvantages of centralized network topology such single point of failure, censorship and scalability. Structured P2P networks fundamentally offer only storage/retrieve operations based on single data items, which are suitable for simple applications such in file sharing. On the other hand, advanced applications, such as online social networks, that run on P2P networks would benefit greatly from more advanced data structures such as lists, sets or trees.

In such applications, every user has a profile with a list of friends, articles with nested comments, a box of messages, and many albums of images that contain a different number of images, in this case, if all of these data elements are stored in a single node, this leads to two things, first, the replication for all data elements would take quite a long time in order to replicate to another node in the network, second, the single storing node itself would be overloaded.

This points to the impracticality of the structured P2P networks for storing such types of data. The objective of this thesis is to achieve a secure Distributed Data Structure (DDS) scheme for use in structured P2P networks. The DDS scheme is designed on the top of DHTs as a storage layer for P2P networks which allows to distribute the load by distributing storage of complex data structure as well as a guarantee that the data which is stored in this scheme is kept secure. This thesis provides a threefold solution. First, we propose the DDS scheme, a distributed storage scheme which supports advanced data structures such as sets, list, and trees as well as guaranteeing the data availability via an appropriate replication protocol. Second, the data in P2P networks is stored among the peers, this mean P2P networks have no single query point. Therefore, in order to add new functionality to the DDS scheme, we add two essential services to the DDS scheme such as metadata search and computational data elements. Finally, we evaluate the security requirements for the DDS scheme in P2P networks in order to ensure the data will still be secure in P2P networks.

5.1 Conclusion

In this section, we conclude the work of our papers, our contributions regarding the secure DDS scheme in P2P networks, discuss different challenges one after the other and we conclude our solutions and approaches regarding these challenges in this thesis:

- **Challenge 1: Storage complex and big data in P2P networks**

  For our first problem statement named Challenge 1, we propose in [3] a distributed storage scheme named DDS which supports advanced data structures such as sets, lists and trees in structured P2P networks as in DHTs. Challenge 1 deals with the limitations of storing complex and big data structures in structured P2P networks, because structured P2P networks offer only storage and retrieval operations based on single data items, in spite of advanced applications, such as online social networks, that would benefit from more advanced distributed data structures such as lists, sets or trees. Therefore, we introduce the DDS scheme to overcome the limitations of using sophisticated applications especially storage dependent applications such as online social networks in P2P networks. The DDS scheme consisted of linked data items. This DDS is basically a pointer graph in the DHT in which each linked data object contains ID information, pointers, and a payload. We show DDS operations as well as individual operations on individual items in the DDS, such as selective modification. In our approach we overcome the storage overloaded by distributing the data among the nodes. In our paper [3], we evaluate the performance and costs for using DDS in small and large scale scenarios, with and without churn, as well as on deep and broad DDSs in a network with 1000 nodes. The results show that the challenge is solved and all elements of each distributed data structure is successfully stored and retrieved and that the approach comes with low overhead and delay. On the other hand, the results of deep DDS show that it takes a longer time for the retrieval process of the data elements, because of the nature of the DDS scheme as a pointers graph, and each item only resolves the pointer to the next element. This leading to a chain of individual item retrievals and for each retrieved DDS scheme object a further DHT lookup is required. The resulting is an increase of the duration times of collective-retrieves for all data elements with the greater depth of DDS due to the nature of DHTs. Therefore, we will deal with this issue in the next challenge.

- **Challenge 2: Accelerate the retrieval process in the DDS scheme**

  For Challenge 2, we introduce and motivate the idea of a Structure Cache, we propose our solution in [6] for improving the collective retrieval time for deep DDS scheme which we presented in [3]. Challenge 2 deals with the time-consuming retrieval of deep DDS scheme which we mentioned in Challenge 1. In our paper [6], we add the Structure Cache to the DDS scheme. In this Structure Cache, we keep and index the overlay keys of the DDS elements at the root of the DDS, so the Structure Cache includes the overlay keys of all deeper DDS objects. This enabled the client node to perform a complete retrieval of the DDS significantly faster via parallel retrievals. We examined the performance of the Structure Cache applied in the DDS under churn and without churn in a network with 1000 nodes. We show in the evaluation results that the DDS operations with Structure Cache are fully functional and resolve to nearly 100% of the retrieve operations as well as the retrieval time of DDS scheme with Structure Cache becomes much faster. The Structure Cache is needed to keep up-to-date when data is added or changed in the structure. Therefore in our paper [1], we focus on the various possible approaches for updating the Structure Caches. We introduce first a Periodic Top-Down Update, using just the minimal information about the structures that are used by the DDT. Then, we introduce an Instant Bottom-Up Update approach, using additional information about the predecessors in the DDT to backtrack the DDT to its root, updating all Caches on its way. We evaluate both approaches in our paper, compare both solutions, and present the advantages and disadvantages of both update mechanisms. In addition, we show that both approaches could be the right choice for an application, depending on how an application uses the Structure Cache and what the main purpose of the application is. As a conclusion for our solutions in [6] and [1], Challenge 2 is solved.

- **Challenge 3/Challenge 4: Services for DDS scheme in P2P networks**

  In the third and fourth problem statement we referred to essential services for DDS scheme which are concluded in the following.

  Challenge 3 addresses the metadata-based search in P2P networks since the DDS scheme in [3] needs suitable search services. Structured P2P networks support a lookup mechanism which uses a simple key-value which is used to lookup for a file or document with a specific key in a fixed time as shown to take O(log N) hops in the overlay network. On the other hand, these networks lack standard algorithms to perform the metadata search. Metadata such as information or attributes related to a specific file for instance name of the author or book title. Therefore, we proposed in [7] algorithms that support metadata searches in structured P2P networks as in DHT. The DDS scheme [3] is introduced to the P2P networks as storage layer that support complex data structures and without appropriate metadata search mechanisms, the functionality of DDS scheme be incomplete. In [7], several search algorithms are introduced to DDS scheme in order to add new functionality to the DDS by adding the service of searching for data items based on their metadata. We propose four search algorithms named LocalJoin, ParallelJoin, NetworkJoin and BloomJoin that can be used to perform the metadata search depending on how the node joins the network and how the search is performed. The performance characteristics of these search algorithms are introduced through simulation. In evaluating the quality and costs of the search mechanisms in DDS, six metrics are used. In order to gain a wider range of insights in the behavior of the search algorithms, three DDS trees are implement and evaluate, the binary tree, deep tree and customized broad tree. The evaluation is done using two search techniques which proceeded the join algorithms, that is, Exhaustive search where the algorithm proceeds through the entire DDS ensuring that it exhaustively searches every node, and First Match search in which the algorithm terminates the search process once it identifies that a match has occurred. Results summary showed that the challenge is solved and the best algorithm with regards to all specified evaluation metrics is the LocalJoin algorithm. In addition, the results show that all algorithms failed to terminate with a single level broad tree with 1000 child nodes, which models a set, three out of four algorithms failed to terminate, LocalJoin is the only exception.

  Challenge 4 is regarding the adding of a new service to the DDS scheme [3], therefore, we introduce the computational elements in our paper [2] which are used widely in sophistical applications for complex event processing and distributing of computational data. These add new functionality to the DDS scheme [3] and allow for ease in extending the use cases of DDS in distributed applications. We introduce an extension for DDS scheme called CDE which interprets and computes the DDS payload. The CDE extension includes an additional backward connection in DDS scheme in order to make the payload up-to-date with the correct function as shown in the following example. In this example when the the value stored under the ID A changes from 2 to 3, then the value of payload also should change, because of the payload is the sum of following arguments.

$$Payload = ValueA + ValueB + ValueC + ValueD \tag{5.1}$$

This is done by implementing four distribution methods called: push without timer, push with timer, pull without timer and pull with timer. Furthermore, the special case of many functions where each function is influenced by another function (chain of functions) is highlighted and discussed. We focus on three main metrics in the evaluation of the quality and cost of distributed computing with DDS named correctness ratio, sum of messages out and sum of initiated lookups. In the evaluations the correctness ratio for the case of push without a timer is the highest at approximately 95%, while the correctness ratio for the case of the pull without timer simulations is approximately 30%. The study shows that our computational data elements with DDS approaches solve Challenge 4. In addition, our approach can benefit the creation of further applications as in complex event processing.

- **Challenge 5: Security and access control for DDS scheme in P2P networks**

  Challenge 5 focuses on security issues for the DDS scheme. Therefore, in our paper [5] we propose and evaluate a security model for DDS scheme based on strictly defined security requirements in details. The DDS scheme builds on the top of P2P networks as a storage scheme. P2P networks are open networks which mean every node/peer can join and participate to the network even malicious nodes. Therefore, the storage of sensitive data in the DDS scheme without security mechanisms would not be secure as there is no guarantee that the data would still remain intact without any modifications, deletions or misuse. In [5], we propose a security model and discuss a broad range of security problems in P2P networks that can affect the DDS scheme. We firstly identify the functional requirements such as confidentiality, integrity and authentication of the data and give the assumptions, such as, the possibility that each participant is potentially a malicious node for the secure model. With these requirements, we determine what features the security architecture must offer. We propose a hybrid cryptography paradigm as a basis for the security architecture design. It is a crucial security requirement to not allow third parties nodes in the DHT. We propose in the secure model an entity named Security Context attached to a DDS which represents the main security module encapsulating the security logic of a DDS. Experimental evaluations indicated that the security architecture with the given feature set, introduces linear time and space overhead while staying scalable for large P2P applications. Thereafter, we propose in [4] our access control architecture for the DDS scheme. The main goal of this paper is to deal with as well as find solutions to the access control problems for the DDS scheme. This work defines the Simple Security Context model as a basic Security Context that consists of three sub-modules: Access Control Map, Secured Data Items and Configuration. We analyze many access problems such as the revocation problem of shared access to secured distributed system. As a summary, Challenge 5 is solved by the contributions in our papers[5] and [4].

- **Challenge 6: Sophisticated application for DDS scheme in P2P networks**

  The last challenge in this thesis is Challenge 6 which deals with using the DDS scheme in real sophisticated application. We propose in our paper [32] a method for the replication of the data in advanced applications such as in online social networks. We show the possibility of replicating and increasing the availability of the content of participants by utilize the privacy policies of the participants. This is done by replication of the content of participants to participants who trust each other without encrypting it, for example the peers belonging to participants who can access the content of user X according to $X'S$ privacy policy. We conduct several experiments through simulation and we consider five reference policies which depend on friendship relationship between the participants. The simulation results show that our approach increases the availability of the profiles' data and contents in the DOSN. We show that Challenge 6 is solved in our paper [32]. Our approach is limited to be used between the peers who trust and know each other in the network, while this approach does not consider if the peers do not know each other, this issue is still open and needs to deals with in future works.

5.2 Future Work

In this part of the thesis, we discuss the remaining points which are raised up during the investigation of the challenges and which we do not cover all of these issues in one thesis. Therefore, in this section, we introduce these points and suggestions as possible future works, with the knowledge that some of these points are mentioned in our papers.

5.2.1 New structures of DDS scheme.

In this thesis, we design a new storage layer for structured P2P networks which allows the storage of complex data structures such as sets, list, and trees as shown in Chapter 2, and mainly, we concentrate our implementations, services and security mechanisms on the distributed data tree, while other data structures such as undirected data graphs where the edges have no orientation, are remaining without implementation. Therefore, the implementation and analysis of the remaining structures of DDS would be one of the future works.

5.2.2 Structure Cache for the DDS scheme.

We present in Chapter 2.2 the proposal of a Structure Cache which is attached to the DDS in order to improve the performance of the data retrieval by maintaining and index data structure at the root of the DDS. This idea can be improved by having many Structure Caches which differ from each other by the storage time and find methods to keep these different Structure Caches synchronized in order to deal with very complex DDS which contains many roots.

5.2.3 Services to the DDS scheme.

The DDS scheme is a storage layer build on the top of DHTs, this scheme gets benefits from the functions that are offered in the lower layers and at the same time generate services in order to add new functionality and improve the quality of the DDS applications. We propose in Chapter 3 two essential services to the DDS scheme called computational elements and search mechanisms and in the future other services such as statistics can be investigated in order to add new functionality to the DDS scheme.

5.2.4 Search metadata mechanisms to DDS scheme.

We present in 3.1 four algorithms which address the support for metadata search in order to add new functionality of DDS scheme. In the future, we need to identify new search algorithms as well as improve the current algorithms by dealing with the problems which appeared during the evaluation as for example all algorithms except LocalJoin failed to terminate in some experiments, in particular, a single level broad tree with 1000 child nodes. Another improvement would be to improve the NetworkJoin algorithm by limiting the number of forwarded messages through checking the recipient of the query messages. Another suggested improvement is the adjustment of the timer mechanism to make more specific statements about the termination result of the algorithm.

5.2.5 Security of the DDS scheme.

Security is a wide field for research because it is an art of mind. In our papers [5] and [4], we propose and evaluate a flexible security model for the DDS scheme based on strictly defined security requirements. With this model, we can guarantees security and access control of the data storage in the distributed data structure within the P2P network using security mechanisms. This model needs more investigation to see how the security mechanisms which are in this model secure under different kinds of attacks are. For example, regarding the Access Control Map which we presented in 4.2. This Access Control Map needs more investigation in order to see how it can be secured against hackers and ensure for example the entries in the Access Control Map cannot be changed.

5.2.6 Data allocation strategies

Back to our a content allocation strategy which we present in [32]. This strategy is supporting the contents availability of participants in online social networks based on the privacy policies of the participants. This approach is limited to be used by participants who have a kind of relationship and trust each other, our approach exploits these relationships to replicate the content between these participants without needing to encryption. Therefore, one of the possible future work is to develop this strategy in order to replicate the content on the networks even if there are no trusted peers available.

5.3 Closing Words

Nowadays, many users and applications shift to use decentralized systems especially after the Snowden scandal, Arab spring and last but not least misuse the privacy such as in cambridge analytica and facebook scandal because of the advantages such as self-organism, and have no singular entity. With the development of decentralization, peer-to-peer alternative models are becoming an increasingly likely replacement for various aspects of internet infrastructure, take as examples of distributed online social networks, distributed computing, blockchain, and cryptocurrency which have given more responsibility, authority and power to users because of it hosting applications and services in a decentralized environment.

Currently, it is obvious to see that the future trend of super companies besides its fields such as Apple shift toward content marketing which is based on receiving money for selling the content. The new trends of content marketing are the decentralized via a P2P or BitTorrent networks without spying, misuse or monetizing that data due to the central access of the provider or central authority responsible for handling users data. Therefore, distributed data structure and decentralization of data is the important task for the meantime

In this dissertation, we focus on the distributed and centralization of data and the improvement of storage approaches in P2P networks and we propose new storage layer in P2P called DDS scheme. Furthermore, the services such as computational elements, search mechanisms as well as the security mechanisms which are proposed in this work are designed to keep the flexible architecture and can be used not only for DDS scheme, therefore, we believe that our contributions and approach in this dissertation can be helpful for further research and re-used to other P2P applications or different distributed networks.

Bibliography

[1] Raed Al-Aaridh, Niklas Foerst, and Kalman Graffi. “Optimization of the structure cache for distributed data structures in overlay networks”. In: *International Conference on Electrical and Computing Technologies and Applications, ICECTA*. 2017, pp. 1–5 (Pages: 7, 12, 13, 19, 20, 25, 26, 30, 33, 36, 44, 45).

[2] Raed Al-Aaridhi, Felix Bandosz, and Kalman Graffi. “Computational Elements For Distributed Data structures On Overlay Networks”. In: *Academics World International Conference AWIC*. 2017, pp. 1–6 (Pages: 8, 12, 25, 29, 45).

[3] Raed Al-Aaridhi and Kalman Graffi. “Sets, lists and trees: Distributed data structures on distributed hash tables”. In: *35th IEEE International Performance Computing and Communications Conference, IPCCC*. 2016, pp. 1–8 (Pages: 6–10, 12, 13, 16, 20, 25, 26, 30–34, 36, 40, 44, 45).

[4] Raed Al-Aaridhi, Ahmed Yuksektepe, and Kalman Graffi. “Access control for secure distributed data structures in Distributed Hash Tables”. In: *IEEE International Symposium on Local and Metropolitan Area Networks, LANMAN*. 2017, pp. 1–3 (Pages: 10, 12, 36, 46, 48).

[5] Raed Al-Aaridhi, Ahmed Yuksektepe, and Kalman Graffi. “Secure model for distributed data structures on distributed hash tables”. In: *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT*. 2017, pp. 1–8 (Pages: 9, 10, 12, 33, 34, 37–39, 46, 48).

[6] Raed Al-Aaridhi, Ahmet Yuksektepe, Tobias Amft, and Kalman Graffi. “Distributed data structures improvement for collective retrieval time”. In: *19th International Symposium on Wireless Personal Multimedia Communications, WPMC*. 2016, pp. 85–90 (Pages: 7, 12, 13, 16, 20, 25, 26, 30, 33, 36, 44, 45).

[7] Raed Al-Aaridhi, Iakov Dlikman, Newton Masinde, and Kalman Graffi. “Search Algorithms for Distributed Data Structures in P2P Networks”. In: *IEEE International Symposium on Networks, Computers and Communications, ISNCC*. 2018, pp. 1–8 (Pages: 8, 12, 25, 26, 45).

[8] 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 (Page: 2).

[9] 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 (Page: 2).

[10] 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 (Page: 2).

[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 (Page: 2).

[12] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S Wallach. "Secure routing for structured peer-to-peer overlay networks". In: *ACM SIGOPS Operating Systems Review* 36.SI (2002), pp. 299–314 (Page: 2).

[13] Mary Subaja Christo and S Meenakshi. "Enhancing security properties of Rumor Riding protocol under various attacks scenario in P2P network". In: *Communication and Signal Processing (ICCSP), 2016 International Conference on*. IEEE. 2016, pp. 1130–1135 (Page: 2).

[14] 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: 2).

[15] 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: 2).

[16] 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: 14).

[17] Matthias Feldotto and Kalman Graffi. "Comparative Evaluation of Peer-to-Peer Systems Using PeerfactSim.KOM". In: *IEEE HPCS'13: Proceedings of the International Conference on High Performance Computing and Simulation*. 2013 (Page: 17).

[18] Jan Graba. "Remote Method Invocation (RMI)". In: *An Introduction to Network Programming with Java* (2007), pp. 136–157 (Page: 27).

[19] K. Graffi, C. Gross, D. Stingl, D. Hartung, A. Kovacevic, and R. Steinmetz. "LifeSocial.KOM: A Secure and P2P-based Solution for Online Social Networks". In: *Proc. of IEEE Int. Consumer Communications and Networking Conf. (CCNC)*. 2011 (Page: 2).

[20] Kálmán Graffi. "Monitoring and Management of Peer-to-Peer Systems". PhD thesis. 2010, pp. 1–295. ISBN: 978-3-86853-658-4 (Page: 2).

[21] Kalman Graffi. "PeerfactSim.KOM: A P2P System Simulator - Experiences and Lessons Learned". In: *IEEE P2P '11: Proceedings of the International Conference on Peer-to-Peer Computing*. 2011 (Page: 17).

[22] 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 (Page: 2).

[23] Kalman Graffi, Christian Gross, Patrick Mukherjee, Aleksandra Kovacevic, and Ralf Steinmetz. "LifeSocial.KOM: A P2P-Based Platform for Secure Online Social Networks". In: *Proc. of IEEE Int. Conf. on Peer-to-Peer Computing (P2P)*. 2010 (Page: 2).

[24] Kalman Graffi, Dominik Stingl, Julius Rückert, and Aleksandra Kovacevic. "Monitoring and Management of Structured Peer-to-Peer Systems". In: *IEEE P2P '09: Proceedings of the International Conference on Peer-to-Peer Computing*. 2009 (Page: 2).

[25] Aleksandra Kovacevic, Sebastian Kaune, Hans Heckel, Andre Mink, Kalman Graffi, Oliver Heckmann, and Ralf Steinmetz. *PeerfactSim.KOM - A Simulator for Large-Scale Peer-to-Peer Networks*. Technical Report: Tr-2006-06. TU Darmstadt, 2006 (Page: 17).

[26] Yi Ma and Dongqi Wang. “A novel trust model for P2P networks”. In: *Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on*. IEEE. 2016, pp. 1969–1973 (Page: 2).

[27] 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).

[28] Michael Backes, Christian Cachin, and Alina Oprea. “Secure key-updating for lazy revocation”. In: *Pearson Deutschland GmbH*. 2006 (Page: 37).

[29] Rajeev R Raje, Joseph I Williams, and Michael Boyles. “Asynchronous remote method invocation (ARMI) mechanism for Java”. In: *Concurrency: Practice and Experience* 9.11 (1997), pp. 1207–1211 (Page: 27).

[30] Norma RFC4949. “Internet Security Glossary, Version 2”. In: *Recuperado el 24 de febrero de 2010*. 2007 (Page: 37).

[31] 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 (Page: 2).

[32] Andrea De Salve, Paolo Mori, Laura Ricci, Raed Al-Aaridhi, and Kalman Graffi. “Privacy-Preserving Data Allocation in Decentralized Online Social Networks”. In: *Distributed Applications and Interoperable Systems - 16th IFIP WG 6.1 International Conference, DAIS 2016, Held as Part of the 11th International Federated Conference on Distributed Computing Techniques*. 2016, pp. 47–60 (Pages: 10, 12, 39, 46–48).

[33] 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).

[34] 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: 2, 13, 25, 26, 29, 32).

[35] Mika Suvanto. “Privacy in peer-to-peer networks”. In: *Helsinki University of Technology T-110.551 Seminar on Internetworking*. 2005 (Page: 2).

[36] Jochem Van Vroonhoven. “Peer to peer security”. In: *4th Twente Student Conference on IT*. Citeseer. 2006 (Page: 34).

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

Raed Al-Aaridhi

Please add here

the DVD holding sheet

This DVD contains:

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