UNDERSTANDING FILE AND INFORMATION SHARING SERVICES IN WEB 2.0.

Demetris Antoniades

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

in Computer Science

in the Graduate Division

of the University of Crete

Heraklion, December 2011

Understanding File and Information Sharing Services in Web 2.0.

A dissertation submitted by
Demetris Antoniades
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate Division of the University of Crete

The dissertation of Demetris Antoniades is approved:

Committee:

Evangelos P. Markatos
Professor, University of Crete – Thesis Advisor

Maria Papadopouli
Assistant Professor, University of Crete

Sotiris Ioannidis
Principal Researcher, FORTH-ICS

Constantine Dovrolis
Associate Professor, College of Computing, Georgia Tech

Thomas Karagiannis
Associate Researcher, Mircrosoft Research

Georgos Siganos
Associate Researcher , Telefonica I+D

Dimitris Nikolopoulos
Associate Professor, University of Crete

Department:

Panos Trahanias
Professor, University of Crete – Chairman of the Department

Heraklion, December 2011

Abstract

Web 2.0 reflects the current practices in web page design and usage that turned the web into an application platform and enabled users to create social communities. In this dissertation we examine two widely used mechanisms that enable file and information sharing in web 2.0. Furthermore, we illustrate the design and applicability of a new technique for performing distributed measurement and monitoring tasks.

First, we look at One-Click Hosting (OCH) services, a relatively new type of services, such as RapidShare and MegaUpload, which offer users the ability to share files through centralized servers. OCH services are continuously increasing and challenge the dominant usage of peer-to-peer (p2p) systems for file sharing. Our study uses a combination of passive and active measurements, and tries to understand OCH service's architecture, usage patterns and content characteristics. Furthermore, we compare RapidShare, the leading OCH platform, with BitTorrent in terms of user-perceived throughput and content availability, and explore the characteristics of some popular RapidShare indexing sites, a fundamental component for content discovery.

Second, we provide a wide characterization study of URL shortening services, a number of services, especially popular within online social networks, that keep a mapping of submitted URLs to shorter equivalents and redirect users to the original URL upon request. Using traces of short URLs collected both from crawling a number of URL shortening services, and collecting short URLs that appear in Twitter messages, we analyze the usage of short URLs in general and in the context of certain social communities. Furthermore, we challenge the performance benefits and implications of their wide usage.

Finally, we present MOR, a technique for performing distributed measurement and mon-

itoring tasks using the geographically diverse infrastructure of the Tor anonymizing network. MOR evolved during our study of One-Click Hosting services and the need for examining their infrastructure properties. Not limited to that analysis, we also provide a number of case studies that show the applicability and value of MOR in examining Internet services and detecting network neutrality violations.

The results of this research are the following. First, we show that OCH services have all the means to compete with p2p systems for becoming the dominant file sharing platform. While built on a typical web data-center infrastructure, their usage patterns suggest little benefit from deploying a more distributed infrastructure. Second, through the study of URL shortening services we observe that short URLs reflect an "alternative" web, with different popularity and lifetime patterns, compared to access patterns as seen for the traditional web community. Overall, our work highlights that if both these services continue to increase they will become part of the web's critical infrastructure with profound impact in it's usage and performance, creating a need for the study and development of alternative architectures and distribution methods.

Thesis Advisor: Prof. Evangelos Markatos

Περίληψη

Ο όρος web 2.0 αντικατοπτρίζει τις τρέχουσες πρακτικές στο σχεδιασμό και τη χρήση ιστοσελίδων, οι οποίες οδήγησαν στη μετατροπή του Παγκόσμιου Ιστού σε πλατφόρμα εφαρμογών και επέτρεψαν στους χρήστες του να δημιουργήσουν κοινωνικά δίκτυα και ομάδες. Στη διατριβή αυτή εξετάζουμε δυο ευρέως διαδεδομένους μηχανισμούς που επιτρέπουν το διαμοιρασμό αρχείων και πληροφορίας στο web 2.0. Επιπλέον, παρουσιάζουμε το σχεδιασμό και την εφαρμοσιμότητα μιας νέας τεχνικής που δίνει τη δυνατότητα για την εκτέλεση κατανεμημένων μετρήσεών και διαδικασιών επίβλεψης δικτύων.

Αρχικά, εξετάζουμε τις υπηρεσίες One-Click Hosting (OCH), ένα σχετικά νέο τύπο υπηρεσιών οι οποίες προσφέρουν στους χρήστες τη δυνατότητα να μοιράζονται αρχεία μέσω ενός κεντρικοποιημένου συστήματος εξυπηρετητών. Οι χρήση τέτοιων υπηρεσιών αυξάνει συνεχώς τα τελευταία χρόνια και τις έχει φέρει σε θέση να απειλούν την πρωτοκαθεδρία των συστημάτων peer-to-peer (p2p) στο τομέα του διαμοιρασμού αρχείων μεταξύ των χρηστών του Διαδικτύου. Στη μελέτη μας, μέσω ενός συνδυασμού από παθητικές και ενεργητικές μετρήσεις, προσπαθούμε να κατανοήσουμε την αρχιτεκτονική των OCH υπηρεσιών, τα πρότυπα χρήσης τους και τα χαρακτηριστικά του περιεχομένου το οποίο ανταλλάζεται μέσω αυτών. Επιπλέον, συγκρίνουμε τις ταχύτητες μετάδοσης και τη διαθεσιμότητα περιεχομένου, στο RapidShare, την δημοφιλέστερη OCH υπηρεσία, όπως αυτές τις αντιλαμβάνονται οι χρήστες του με τις αντίστοιχες που αντιλαμβάνονται οι χρήστες του ΒιτΤορρεντ. Τέλος, διερευνούμε τα χαρακτηριστικά μερικών δημοφιλών ιστοσελίδων-ευρετήρια, οι οποίες αποτελούν ένα θεμελιώδες κομμάτι της διαδικασίας ανακάλυψης περιεχομένου στο RapidShare.

Στη συνέχεια προχωράμε σε μια μελέτη κατανόησης των υπηρεσιών URL shortening, ένα είδος υπηρεσιών, ευρέως διαδεδομένα στις υπηρεσίες κοινωνικής δικτύωσης, οι οποίες κρατάνε την

αντιστοίχηση ενός URL σε ένα μικρότερο ισοδύναμο και ανακατευθύνουν τους χρηστές στο αρχικό URL όταν αυτοί το ζητήσουν. Χρησιμοποιώντας δεδομένα για τα short URLs που μαζέψαμε μέσω επισκέψεων σε ένα αριθμό από URL shortening υπηρεσίες, αλλά και συλλέγοντας short URLs που εμφανίστηκαν σε μηνύματα στο Twitter, καταφέρνουμε να αναλύσουμε τη χρήση των short URLs γενικότερα στο Παγκόσμιο Ιστό, όσο και στο πλαίσιο συγκεκριμένων κοινωνικών ομάδων. Επίσης, μελετάμε τα πλεονεκτήματα της χρήσης των short URLs αλλά και τις επιπτώσεις που προκύπτουν από αυτή.

Τέλος, παρουσιάζουμε το MOR, μια τεχνική που δίνει τι δυνατότητα για την εκτέλεση κατανεμημένων μετρήσεών και διαδικασιών επίβλεψης δικτύων χρησιμοποιώντας τη γεωγραφικώς κατανεμημένη υποδομή του διχτύου ανώνυμης πρόσβασης, ToR. Το MOR αναπτύχθηκε κατά τη διάρκεια της μελέτης των υπηρεσιών One-Click Hosting, μέσα από την ανάγκη για κατανόηση της υποδομής των υπηρεσιών αυτών. Επιπλέον αυτού, παρουσιάζουμε έναν αριθμό από περιπτώσεις που δείχνουν την εφαρμοσιμότητα και τις δυνατότητες που προσφέρει το MOR στην εξέταση διαφόρων διαδικτυακών υποδομών καθώς και στη ανίχνευση παραβιάσεων της ουδετερότητας του Διαδικτύου.

Τα αποτελέσματα της έρευνας μας μπορούν να συνοψιστούν στα ακόλουθα. Πρώτο, δείχνουμε ότι οι υπηρεσίες OCH έχουν όλες τις προϋποθέσεις για να ανταγωνιστούν τα συστήματα p2p και να γίνουν η κύρια πλατφόρμα διαμοιρασμού αρχείων στο διαδίχτυο. Τα πρότυπα χρήσης τους δείχνουν ότι η ανάπτυξη μιας κατανεμημένης υποδομής δε θα οδηγούσε σε σημαντικά οφέλη, και δικαιώνουν την επιλογή μιας τυπικής web υποδομής. Δεύτερο, μέσω της μελέτης των υπηρεσιών URL shortening παρατηρούμε ότι η χρήση των short URLs αντικατοπτρίζει ένα «εναλλακτικό» ιστό, με διαφορετικά πρότυπα δημοτικότητας και διάρκειας ζωής, σε σχέση με τα αντίστοιχα στο παραδοσιακό Παγκόσμιο Ιστό. Συνολικά, η δουλειά μας τονίζει πως εάν η χρήση των δύο αυτών υπηρεσιών συνεχίζει να αυξάνει τότε οι υπηρεσίες θα γίνουν μέρος της υποδομής ζωτικής σημασίας του Παγκόσμιου Ιστού, με σημαντικές επιπτώσεις τόσο στην χρήση όσο και στην απόδοση του. Οι επιπτώσεις αυτές δημιουργούν την ανάγκη για τη μελέτη και ανάπτυξη εναλλακτικών web αρχιτεκτονικών και μεθόδων διαμοιρασμού.

Επόπτης: Καθηγητής Ευάγγελος Π. Μαρκάτος

The research outlined in this dissertation has been funded by Herakleitos II Ph.D. Scholarship in the area of "Internet traffic Classification". The fund is part of the framework of implementation of the project "Hrakleitos II University of Crete" under the Operational Programme "Education and Lifelong Learning 2007-2013" (E.P.E.D.B.M.) of NSRF (2007-2013), which is funded by the European Union (European Social Fund) and Greek National resources.

The research outlined in this dissertaion has been conducted in the Distributed Computing Systems laboratory, which is hosted in the Institute of Computer Science of Foundation of Research and Technology, Hellas.

Acknowledgments

All these years I had the fortune to be surrounded by a number of wonderful people providing me with strength, inspiration and influential support whenever needed. A big thanks to all of you.

First of all I would like to say a special thanks to my parents, Panikos and Rena, and to my brother, Christodoulos, for their continuous support through my whole life experience.

I would also like to thank my advisor, Evangelos Markatos, for giving me the guidelines through my whole graduate studies, and for putting all those questions in front of me, both academic and not.

Everything is always more enjoyable when you are accompanied by friends, and I was lucky to be spending my hours in the lab surrounded by them. Thanks to all of you guys.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the last years the World Wide Web (WWW), mostly referred to as the web, has emerged as a major application platform. Covering the web 2.0 era, the web has evolved from simple web pages and static images to a number of Rich Internet Applications (RIA). Functionality that is traditionally provided by different applications is now available through the web Browser window. This includes video streaming, IPTV, radio streaming, RSS readers, file sharing through services such as RapidShare, online social networking applications, gaming applications, and last, but not least, office suites and web operating systems providing accessibility from many different locations.

At the same time, a number of emerging changes to the web page structure and a new type of web applications, namely blogs and wikis, transformed the Web to a major social interaction platform for the online world. Allowing users to add content to a web page that they do not control, either in the form of comments or media files, introduced User Generated Content (UGC) to be hosted into web servers. Also, with the appearance of a number of Online Social Networks (OSNs) and accompanied social applications and tools, that enable users to share information through their social graph, interaction between users through the web is continuously approaching real life interactions.

Along the transition of the web to a major application and social platform, new sharing mechanisms and platforms have emerged either to ease the process of file sharing or to enable users to share more information in a more space efficient manner.

1.1 Problem Statement and Approach

Traditionally, one of the most used applications in today's networks is file sharing among Internet users. Different types of files, such as movies and tv-series, music and podcasts, applications and books, are legally or illegally exchanged by Internet users on a daily basis. File sharing evolved from traditional methods like FTP services during the 90s, to a variety of p2p applications at late 90s and until nowadays. During the past years a number of online web hosting services, namely One-Click Hosting (OCH) services, appeared enabling users to upload their files either for backup or for sharing among their friends. Along with the use of a number of social tools, that are utilized to provide indexing to files for the greater public, these services are now responsible for a large percentage of the file sharing network traffic volume [PFA+10a]. For this reason, we argue that understanding these services, their infrastructure organization and their usage patterns would be of great benefit for future file hosting services and also core and last-mile Internet Service Providers (ISP) for managing their network infrastructure.

Web 2.0 also changed the way content and information is spread among users. Indeed, recent real-life events, from earthquakes to celebrity news, showed that news are rapidly distributed through Online Social Networks [KLPM10, LG10]. To this extent, URL shortening evolved into one of the main practices for the easy dissemination and sharing of URLs. Motivated, mainly, from the character limit imposed in micro-blocking services, like Twitter, short URLs are now a typical method for information sharing and content propagation. With their increased usage in OSN communities, short URL services have grown significantly in popularity, that they now account for as much as one percent of the total web population per day [ale]. It is obvious that if this trend continues, URL shortening services will become a critical part of web content access process, posing challenging questions regarding web's performance, scalability and reliability. Additionally, a study of these services provides insight into the interests of the OSN communities and a better understanding of their characteristics.

In this thesis we study in depth two widely deployed mechanisms for file and information sharing. We first examine One-Click Hosting services, a widely-used file sharing infrastructure

that is currently responsible for more than 15% of the total web traffic [PFA+10a]. Our examination uses both passive and active measurements to analyze One-Click Hosting services from multiple perspectives, including their data-center infrastructure decisions and their client usage habits. We compare the efficiency of this last-generation web file sharing method with the currently popular p2p file sharing system, BitTorrent. Furthermore, we look at a major sharing component, closely attached to OCH services, namely content indexing web sites, aiming to characterize the type of content users share and whether that content is copyrighted.

We, then, move on to study Short URL services, a widely-used information sharing method in OSNs. We look at the usage and publishing habits of Short URL clients and also evaluate the effect that they have on the web infrastructure. Our study is based on traces of short URLs as seen from two different perspectives: i) collected through a large-scale crawl of URL shortening services, and ii) collected by crawling Twitter messages. The former provides a general characterization on the usage of short URLs, while the latter provides a more focused view on how certain communities use shortening services. Our analysis highlights that domain and website popularity, as seen from short URLs, significantly differs from the popularity distributions provided by well publicized services, such as Alexa. The set of most popular websites pointed to by short URLs appears stable over time, despite the fact that short URLs have a limited high popularity lifetime. Surprisingly short URLs are not ephemeral, as a significant fraction, roughly 50%, appears active for more than three months. Overall, our study emphasizes the fact that short URLs reflect an "alternative" web and, hence, provide an additional view on web usage and content consumption complementing traditional measurement sources. Furthermore, our study reveals the need for alternative shortening architectures that will eliminate the non-negligible performance penalty imposed by today's shortening services.

We proceed with the presentation of *MOR*, a technique for performing distributed measurement and monitoring tasks using the diverse infrastructure of the Tor anonymizing network. [DMS04] MOR emerged from the need to understand different practices when examining the infrastructure of the aforementioned OCH services, where it was extremely valuable in performing distributed measurements using a number of different Internet Service Providers

(ISPs). Not limited to this, we also present a number of case studies where MOR is used in order to examine Internet services and also detect a number of network neutrality violations.

Finally, we present a technique for the identification of user sessions in the Web 2.0 era. As discussed Web 2.0 has affected the way users interact with web-pages, increasing the time a user spends at a page and also the number of actions it does on it. Traditional methods for identifying Web Sessions collect bunchs of HTTP(s) request-response pairs separated by the estimated "think" time of the user. In the Web 2.0 era these methods fail to collect the continuous activity of the Web pages in a single session for each application. In Chapter 6 we define Web Activity Sessions and present a new methodology for their identification. These sessions include the total activity of the user with a Web application/page and give an insight to the behavior of Web users in the Web 2.0 era.

1.2 Novelty and Contributions

In this dissertation we make numerous contributions. More precisely, the contributions are the following:

1. We present the first large scale study of an emerging, web-based, file sharing mechanism, One-Click Hosting services, and analyze their usage, infrastructure and communities. We discuss One-Click Hosting services in detail in Chapter 3.

2. We present the first study of URL shortening services, an important part of micro-blocking services that helps users share more information. As URL shortening services become more popular they also become an important part in the infrastructure and performance of the web. We discuss URL shortening services in more detail in Chapter 4.

3. We present MOR a methodology that enables any user to perform distributed monitoring tasks by using the ToR infrastructure. Using our methodology users and administrators can check for privacy, network neutrality and availability issues for any Internet Service. We discuss MOR in Chapter 5.

1.3 Thesis Statement

Our work highlights the need for the study and development of alternative architectures for file and information sharing. Both the services we study have the potential to grow in such an extent that they become part of the web's critical infrastructure. OCH services can clearly compete with peer-to-peer systems and become a dominant file sharing platform. Their usage patterns suggest that serving content through a typical web data-center infrastructure is not always optimal either for the services themselves or for Internet Service Providers. Additionally, our study on URL shortening services highlights the need for better understanding of the communities created through Online Social Networks, that form an "alternative" web that requires different handling by both content publishers and providers.

1.4 Dissertation Organization

This dissertation is organized as follows. In Chapter 2 a short introduction of historical background knowledge is presented. Our work in One Click Hosting services is presented in Chapter 3. Chapter 4 presents our work in analyzing the usage of Short URL and their hosting services. Chapter 5 presents a technique we developed that assists in performing distributed monitoring using the Tor infrastructure. Chapter 6 presents the design, implementation and evaluation of our Web Session identification technique. Related work is listed in Chapter 7. This dissertation concludes in Chapter 8. A list of publications related to this dissertation's content is presented in Appendix A.

Chapter 2

Background

In this chapter we give a short introduction of some important concepts used in the rest of this dissertation's content.

2.1 Web 2.0

Web 2.0 appeared as a term during the late 90s [DiN99]. Though it does not include any updates to the technical specification of the traditional web, it refers to a number of changes that affect the way web pages are designed and used. The traditional web was considered as a system of interlinked documents where any individual could share her thoughts and interests with all Internet users. Web 2.0 introduced a number of changes that enabled users to also interact with the web applications and with each other by creating communities of same interests without the need for each member to host her individual web page.

In general web 2.0 can be described by three main concepts: [Or07]

- Rich Internet applications that define a new generation of web applications that bring the desktop experience to the browser window, both in terms of graphical representation and usability.
- Service-oriented architecture (SOA) that defines how web 2.0 applications expose their functionality so that other applications can be envinched by leveraging and intergrating their functionality resulting in a richer set of applications.

- Social web that enables users to have greater interaction with the web application, in a way that the user becomes an integral part.

Combining these concepts web 2.0 sites provide users with information storage, creation and dissemination capabilities that were not possible in the traditional web. All these changed not only the way end-users use the web today but also the way the web is organized, where information is stored and how information is propagated between users. Two profound examples of these applications are file sharing through One-Click hosting services and information sharing through URL shortening services. File and information sharing are two widely used applications in todays' Internet. The enhancements introduced to the web provided the means for these applications, and their infrastructure, to become part of the web's critical infrastructure, posing challenging questions regarding its performance, scalability and reliability.

2.2 One-click Hosting File Sharing

With the emergence of the Internet, file and information sharing moved from removable media to online sharing through a large variety of protocols and applications over time. Starting from Usenet [DET80], FTP [PR85] and IRC [OR93] in the 80s, moving on to web indexed FTP services (like mp3.com) during the mid 90s and expanding to the majority of Internet users through p2p systems in the late 90s and further on.

The most widely known and used peer-to-peer system, BitTorrent, is continuously being used by millions of users during the last decade for the exchange of files. During this period file sharing became the most popular Internet user activity and the leading source of network traffic, reaching as high as 60-70% of the total volume in some ISPs [ipob].

With the aforementioned enhancements to the web, a number of web applications gave users the opportunity to upload large files into their servers enabling them to share this content with the broader web community. Sites like RapidShare and MegaUpload enabled end-users to share any type of file by providing storage and download capabilities. This type of services are usually referred to as One-Click Hosting services. Users who upload any file to these services are provided with a URL pointing to it. Using this URL users are then

capable to share their content with their friends, online communities or the broader web. These services have the potential to offer users better performance and availability than p2p systems.

Furthermore, with the emergence of the social web, users index the content using, mainly, blogs and wikis. This enables the creation of a number of communities, either content or location based. These communities play an important role in how the content is then distributed. The exact same content might be shared by a number of communities using different URLs thus limiting the possibilities for ISP level caching or utilizing a Content Distribution Network (CDN).

2.3 URL Shortening Services

Micro-blocking services, like Twitter, are nowadays considered one of the largest news dissemination networks in the online world [KLPM10]. These services, allow users to post small messages that penetrate through a graph of social relationships. Their imposed limit on the number of characters per message, led to the emergence of a new practice for sharing information, URL shortening.

URL shortening services provide users with a smaller equivalent of any provided long URL, and redirect subsequent visitors to the intended source. For example, if a user submits `http://www.this.is.a.long.url.com/indeed.html` to bit.ly, the service will return the following short URL to the user: `http://bit.ly/dv82ka`. This short URL is used exactly as the long URL would be used by it's publisher. The difference comes when the URL is clicked, where a new step is introduced. During this extra step the content consumer is redirected to the URL shortening service in order to get the original URL. Nowadays, URL shortening services are widely used by Twitter users. Indeed, according to alexa.com the most popular URL shortening service serves about 1% of the daily web population. It is more than obvious that if this trend continues, the extra step added from URL shortening services during URL retrieval will affect the performance and scalability of web services, consequently affecting web user's quality of experience.

Chapter 3

One-Click Hosting Services

In this chapter we present a detailed study of One-Click Hosting traffic and services, focusing on the most popular such service: RapidShare. Through a combination of passive and active measurements, we attempt to understand their service architecture, usage patterns, and content characteristics. We also compare RapidShare with BitTorrent in terms of user-perceived throughput and content availability, and we explore the characteristics of some popular RapidShare indexing sites.

3.1 Overview

Over the past decade, file sharing has become one of the most popular Internet user activities, surpassing in terms of traffic volume email, ftp, and even the mighty World Wide Web. Indeed, file sharing accounts for the largest portion of network traffic, reaching as high as 60-70% of the total volume in some ISPs [ipob]. This popularity of file sharing was fueled by the evolution of the p2p paradigm, which enabled users to exchange files without having to rely on a third-party server infrastructure.

As of 2005 a new type of file sharing service has emerged, usually referred to as One-Click Hosting (OCH). OCH sites, such as RapidShare, MegaUpload or FileFactory, allow users to share files through dedicated server infrastructures. Using OCH services, a user can share a large file (even gigabytes) with one or more other users in a sequence of few simple steps:

i. The user uploads the (potentially encrypted) file on an OCH server through a basic web interface.

ii. The OCH service provides the uploader with a URL for the file.

iii. The uploader shares that URL (and a decryption key, if necessary) with other users either privately (e.g. email) or through public indexing sites, such as rslinks.com, blogs, personal web pages, etc.

iv. For publicly indexed content, users can find the download-URL through search engines, and they can then download the file through a basic web interface. Usually, OCH services offer both Free and Premium (i.e. paying) accounts as well as several incentives (discussed later) for attracting users to upload content (making the service more valuable) and to subscribe as Premium users.

Compared to p2p applications, OCH services could provide several advantages for file sharing:

- **Availability:** Files uploaded on OCH services are available 24/7 - not only when a "seeder" is available as is the case with p2p systems.

- **Anonymity:** In the p2p paradigm, peers have to disclose their IP addresses to the community when they are downloading (or uploading) a file. Thus, someone can infiltrate a p2p system and harvest IP addresses of hosts that engage in file sharing. On the contrary, in OCH services the IP addresses (and in some cases the identity) of the uploaders and downloaders are known only to the OCH service. Unless the OCH service cooperates, it would be difficult for someone to harvest IP addresses of computers that engage in file sharing through OCH sites.

- **Performance:** Because of their business model and always-on infrastructure, OCH services aim at offering high throughput, at least to their Premium users. Indeed, our experiments in Section 3.7.1 show that a Premium RapidShare user receives an order of magnitude higher throughput than a BitTorrent user downloading the same file.

- **Content Availability:** As we show in Section 3.7.2, even though a user can find popular content in both OCH and p2p services, less popular/known content can be found more frequently at OCH services.

- **Incentives:** OCH services allow frequent uploaders to receive a higher download throughput, providing the incentive to upload more files, thus making OCH services richer in content.

In this chapter, we study the OCH paradigm from multiple perspectives, focusing on RapidShare, the most popular (at least in terms of traffic) OCH service today [ipob]. Through a combination of passive and active measurements, we analyze traffic characteristics, usage patterns, and especially, we explore their service in terms of location, multihoming, load balancing, file allocation policy, etc. Further, we compare RapidShare and BitTorrent in terms of user-perceived performance and content availability. Finally, we look at the major OCH indexing web sites, aiming to characterize the type of content users share and whether that content is copyrighted.

The contributions of this chapter are:

1. We provide the *first* to our knowledge study of OCH services focusing on their client, server, traffic, content and performance characteristics. This study improves our understanding of OCH services and it allows a comparison with other similar services, such as web Content Distribution Networks (CDN).

2. Using passive network monitoring, we measure the volume of OCH traffic and show that it surpasses (or it is comparable to) the traffic volume of popular video services such as YouTube and GoogleVideo.

3. We show that most files are requested only once in our client-side traces (collected at two academic networks) - very few files were requested more than five times during our 5-month long traces. This suggests that caching, often used in CDNs, may not be effective in OCH services.

4. We explore the number of users that upload content, based on indexing sites, and show that it is only a handful of users that provide most OCH content, making the entire system quite sensitive to minor changes in the number and "productivity" of uploaders.

5. By using the Tor anonymity network as a large geographically distributed client base, we develop a methodology to infer the server location, content replication approach and load balancing that RapidShare uses.

6. We compare the relative benefits of OCH and p2p file sharing systems using RapidShare and BitTorrent, in terms of performance and content availability. We show that RapidShare provides significant benefits in these two aspects.

7. Our analysis of few public indexing sites shows that users often rely on OCH services to share movies, songs, games, and software, and that in most cases this content is protected by copyright legal constraints making such file sharing illegal.

3.2 Description of One-Click Hosting Services

Since early 2005 several OCH sites made their appearance, including `megaupload.com`, `rapidshare.com`, `filefactory.com`, and others. These sites facilitated the creation of a vibrant user community that shares files reliably and inexpensively. OCH services can be described as web services that allow a user to upload and store files on dedicated, always-on servers, and then share those files with other users through a URL (see Figure 3.1). The whole process is usually offered for free and needs a small number of steps for both uploading and downloading files. OCH services do not offer indexing or search capabilities, and search engines cannot crawl the download URLs. What happens, however, is that there are other "indexing" web sites, such as `EgyDown.com` or `RapidShareIndex.com`, in which upload users can post OCH URLs for the files they want to share. Those indexing sites are often publicly visible and crawled by search engines.

In this chapter we focus on the largest and most popular OCH service: RapidShare. RapidShare was launched in October 2006, and within two years it grew to 2.5 million users.[1]

[1] `http://siteanalytics.compete.com`

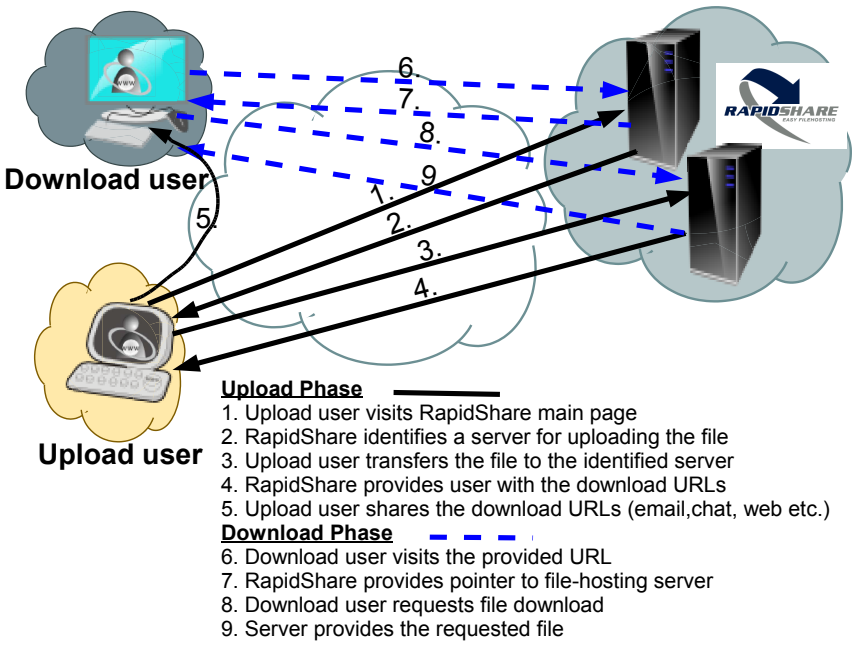


Figure 3.1: Sequence of steps to share a file through RapidShare.

In our client-side traffic traces (§ 3.3), RapidShare is responsible for more than 50% (on average) and up to 95% of the total traffic exchanged through OCH Services. [2]

The main page of RapidShare offers a basic web interface containing just the necessary fields to upload a file. Once a file has been uploaded, RapidShare provides the user with two URLs: the "download URL" that is to be shared with other users who want to access the file, and the "remove URL" to request deletion of that file. To create a viable business model, RapidShare offers two types of service: "Free" and "Premium". **Premium users** can enjoy unlimited use of RapidShare resources: their upload and download bandwidth is not throttled, they can start several concurrent downloads, upload files as large as 2GB, etc. On the other hand, **Free users** are given a download bandwidth that is no more than 200-2000kbps, they can do only one concurrent download at a time, the maximum size of an uploaded file is no more than 200MB (it used to be 100MB until recently), [3] there is a mandatory wait time between successive downloads, etc. To entice users to upload as much content as possible,

[2]Similar results are shown by ipoque's Internet studies [ipoa].

[3]To overcome this file size limit, uploaders partition large objects in files of that size and share them through several URLs.

Name	Collection period	Tot. Bytes	Flows
Monitor1	Jun 6 - Oct 23'08	60.8TB	2.2B
Monitor2	Aug 10 - Dec 2'08	214.8TB	1.4B

Table 3.1: Description of the client-side datasets.

Name	HTTP	BitTorrent	OCH	RapidShare	RapidShare Clients
Monitor1	12.8%	44.5%	3.32%	2.7%	748
Monitor2	4.72%	56.4%	0.23%	0.22%	449

Table 3.2: Distributions of protocols in the client-side datasets.

RapidShare offers "points" to an uploader each time her content is downloaded. Points, can, in turn, be exchanged for Premium accounts or extra download capacity. Such incentives have quickly enabled RapidShare to store a huge variety of songs, movies, games, software, books and other types of content. Last but not least, it is important to note that RapidShare is not responsible for any copyright violations due to illegal file sharing conducted using their infrastructure. Uploaders need to declare that the uploaded content is not protected by copyright laws, and RapidShare is further legally protected by not offering any indexing or search facilities for the content they store.

3.3 Main datasets

Table 3.1 summarizes the main dataset of this study - we refer to this dataset as our "client-side" traces. We collected flow traces in the IPFix format, as well as HTTP packet headers, at two monitoring locations: `Monitor1` is the main Internet access link of a National Research Network that serves a population of about 10,000 students and academics. `Monitor2` is the main Internet access link of a University campus network of about 1,000 students and faculty. Both monitoring points are located in Europe and have a user base composed mostly of university students. One may argue that these demographics are biased and not general in terms of age and location. Our analysis, however, focuses on information that would not depend, most likely, to these demographics. Data from Monitor1 cover about 4 months, while the Monitor2 data cover about 5 months. Table 3.2 breaks down the percentage of bytes that were produced by the applications of interest for each of the datasets. Table 3.2 also mentions

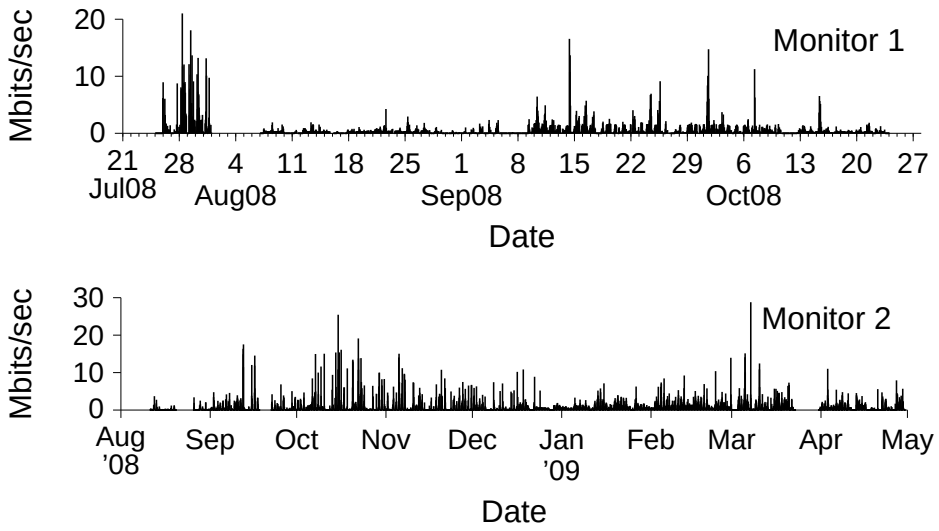


Figure 3.2: Download traffic rate (at the two monitored sites) from all major OCH services in hourly intervals.

the total number of unique client IP addresses that access RapidShare during the monitoring period from each of our monitoring sites (last column). 750 IP addresses have been listed in Monitor1 and 450 IP addresses at Monitor2. [4] These numbers should be viewed as an upper bound on the number of actual RapidShare users in the two monitored networks because some users may be using DHCP. The identification of RapidShare flows was performed using the HOST header field of HTTP requests (searching for the string "rapishare.com"). Based on that data, we then identify all relevant RapidShare flows in our traces. We also use some additional datasets that are described later in the chapter, when first introduced.

3.4 OCH traffic volume

In this section we examine the traffic volume of OCH services, and of RapidShare in particular, in our client-side traces. We compare the traffic volume that these services generate with web and BitTorrent traffic, as well as with major video streaming services.

As shown in Table 3.2, in Monitor1 OCH services generate 3.32% of the total traffic volume. This is low compared to BitTorrent traffic (44.5%) but a significant fraction (25%) of the total HTTP traffic (12.8%). The fraction of OCH traffic is significantly lower in

[4] All IP addresses were anonymized in the traces.

Monitor2. Figure 3.2 shows the aggregate traffic rate downloaded from all OCH services in hourly intervals (the curve for Monitor2 includes 5 more months of data). [5] The hourly rates vary widely and, even though the long-term average is only about 1Mbps in either monitor, the hourly OCH rate often reaches up to 10-20Mbps, which is significant compared to the total traffic rate in these traces. We were hoping to see a clear trend in this timeseries, but this is not the case even in the 9-month period covered in Monitor2.

RapidShare generates more than 80% of the OCH traffic volume in our traces; so, in the rest of this chapter we focus on this particular OCH service. Figure 3.3 compares the average daily download rate of RapidShare with two popular video streaming sites: YouTube (`www.youtube.com`) and GoogleVideo (`googlevideo.com`). Note that RapidShare generates traffic volume which is more than (Monitor1) or comparable to (Monitor2) the traffic volume generated from these major content providers.

Finally, Figure 3.4 shows the hourly traffic rate for web and OCH traffic during a randomly selected week in September 2008 at Monitor1. Note that OCH traffic follows a similar diurnal pattern with web traffic: much less activity during the evening hours and weekends. Afterall, OCH can be viewed as just another web service.

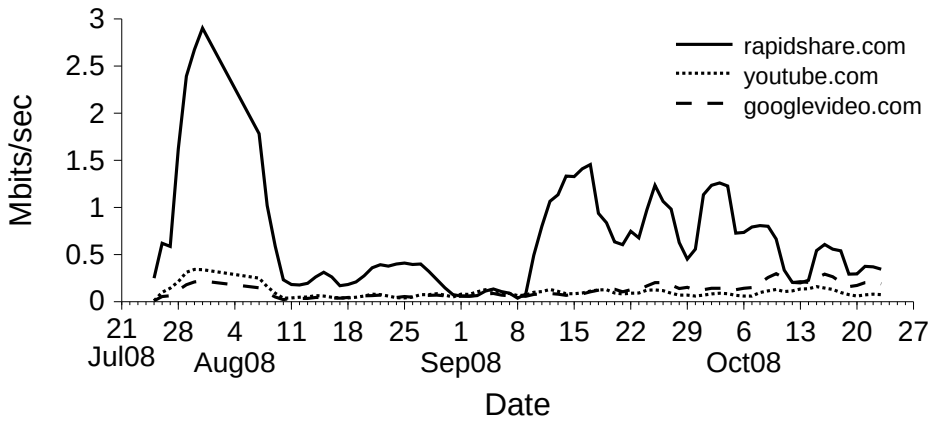
3.5 Characterization of RapidShare clients

In this section, we focus on the characteristics of RapidShare clients that are active in our two client-side traces. As mentioned in Table 3.2, an upper bound for the number of RapidShare clients during the course of our study is around 750 for Monitor1 and 450 for Monitor2.

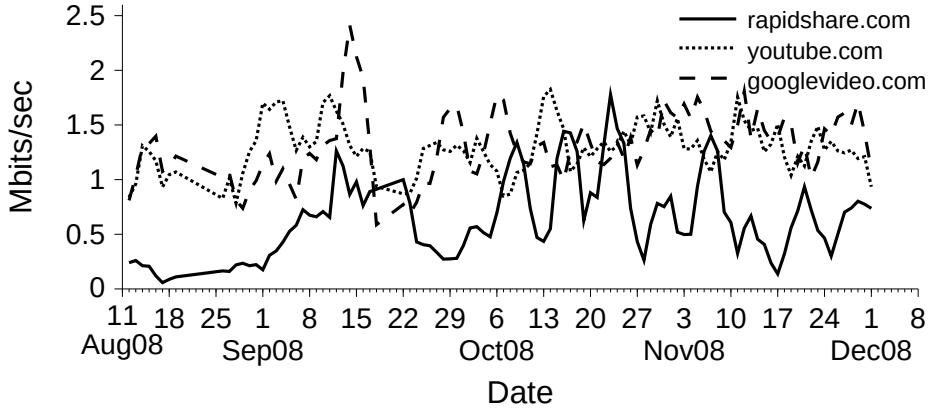
3.5.1 Flow sizes and client downloads

First, we analyze the flow sizes downloaded by RapidShare clients. Figure 3.5 shows the CDF of the download flow sizes for all RapidShare connections. 50% of the flows are smaller than 700 Bytes at Monitor1 and 7 KBytes at Monitor2, while 90% of the flows are smaller than 150KB at both monitors. Since most files provided by RapidShare are typically several megabytes long, these smaller flows probably correspond to web page accesses, failed/stopped

[5] We used the list of OCH services given at `http://en.wikipedia.org/wiki/One-click_hosting`.



(a) Monitor 1



(b) Monitor 2

Figure 3.3: Download traffic rate from RapidShare, YouTube and GoogleVideos in daily intervals.

downloads or web page refreshes to see the remaining wait time until the next download can start. The remaining 10% of the flows, which transfer more than 150KB, are probably actual downloads from RapidShare. In the rest of the chapter we use a threshold of 150KB to distinguish between "content download flows" (larger) and "browsing flows" (smaller). As shown in Figure 3.5, the CDF increases slowly after 100KB or so, meaning that the identification of content download flows should be robust to the selection of this threshold.

Figure 3.6 shows the complementary CDF (C-CDF) of content download flow sizes. Up to the point of 100MB, flow sizes appear to be Pareto distributed, as probably expected. For larger flows we note two significant drops, one at 100MB and another at 200MB. These sizes correspond to the maximum upload file size limits that RapidShare enforces for free

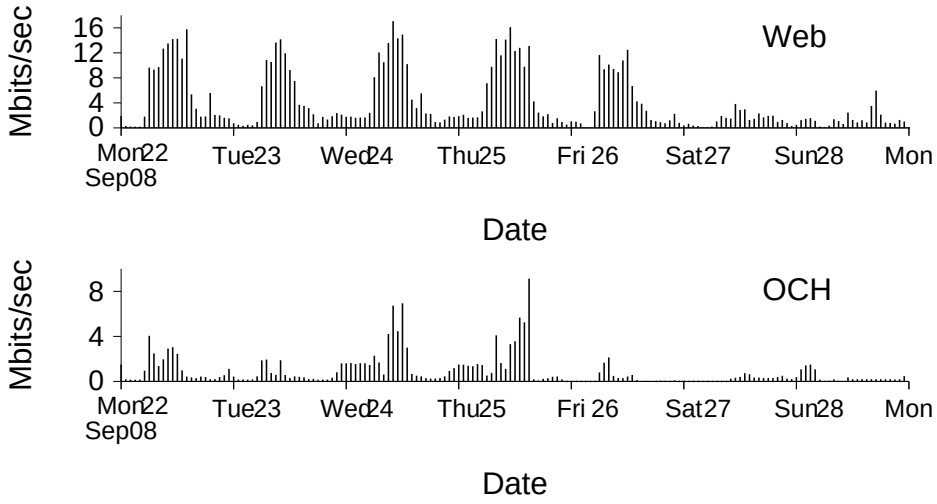


Figure 3.4: Download traffic rate for HTTP and RapidShare in hourly intervals during a week in September'08 (Monitor1).

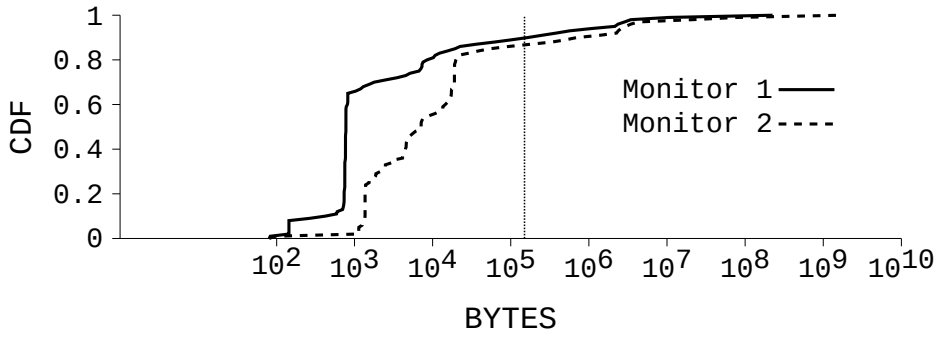


Figure 3.5: CDF of RapidShare download flow sizes at the two monitors.

uploaders: this limit was 100MB and was increased to 200MB in October'08. Premium users, on the other hand, can download and upload files up to 2GB. The difference between the two distributions for files larger than 200MB implies that there are much fewer Premium users at Monitor1 than at Monitor2.

Next, we examine the number of daily content downloads per user (or client). We assume that a client uses the same IP address during the day, and so the daily downloads from the same IP address are interpreted as downloads from the same user. Most clients perform more than one download per day (57% of clients at Monitor2), and only 23% of clients perform more than 10 downloads in the same day. Note that large objects (movies, software, etc) are often split into 100MB or 200MB files, and so a large number of consecutive flows by the same

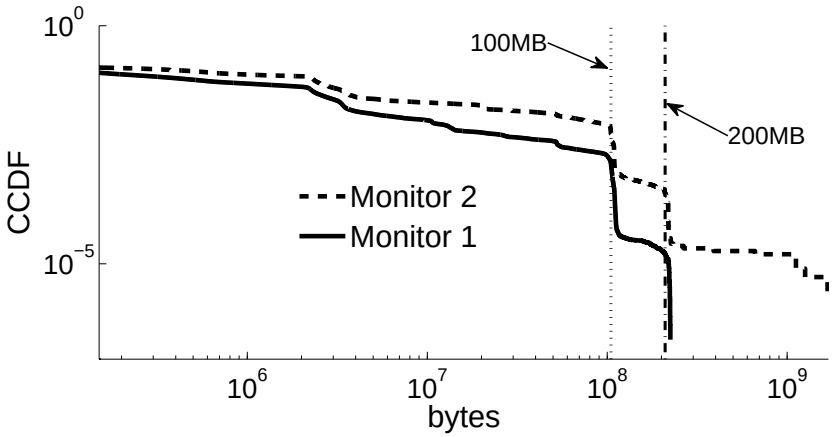


Figure 3.6: C-CDF of RapidShare content download flow sizes at the two monitors.

user may correspond to downloading different parts of the same object. Figure 3.7 shows the C-CDF of the daily number of content download flows per client at Monitor2. This empirical distribution can be approximated by a Pareto distribution with shape parameter 0.66. This low value of the shape parameter implies extremely large variability, to the point that neither the variance nor the mean of the underlying distribution are well-defined. The C-CDF for Monitor1 is similar.

3.5.2 Premium vs. Free users

As previously mentioned, RapidShare supports two user types: **Free** and **Premium**. All content is available to both types and the main difference is that the former are limited in terms of their upload flow sizes, concurrent downloads, and download throughput. RapidShare reports that the download throughput of Free users is throttled to 200-2000kbps. In this section we attempt to identify Premium users based on their download throughput, assuming that content download flows that receive more than 2Mbps are generated by Premium users. Of course, we may underestimate the number of such users when their throughput is limited by their access link capacity or by Internet congestion, and not by RapidShare.

Figure 3.8 shows the distribution of the average download rate per user observed on a daily

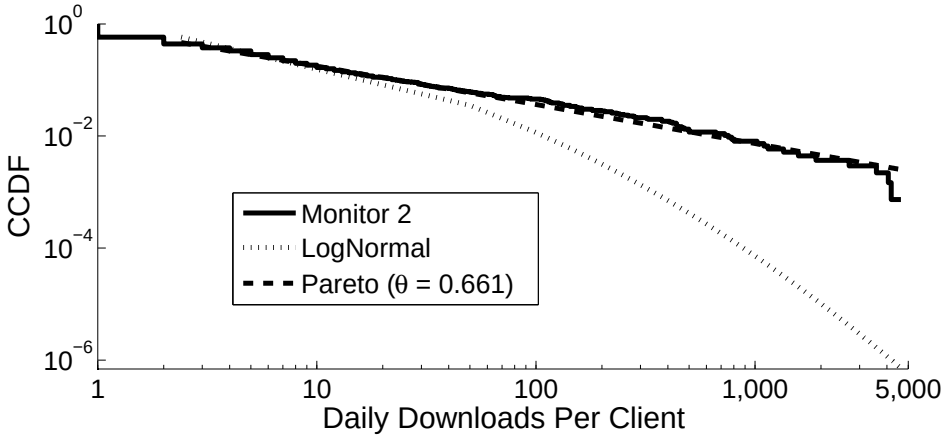


Figure 3.7: Distribution of daily content downloads per client at Monitor2.

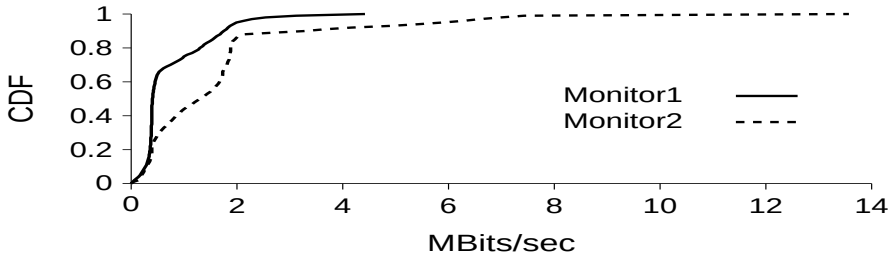


Figure 3.8: Distribution of the average content download throughput per user, measured on a daily basis.

basis. The rates were calculated considering only content download flows. Almost all users at Monitor1 (95%) and more than 80% of users at Monitor2 experience an average throughput that is less than 2Mbps, suggesting that most of them are Free users. At Monitor2, the remaining 20% (5% for Monitor1) of the users enjoy throughput up to 14Mbps and they must be Premium customers. Though, the percentage of Premium users is not very large, and may vary based on the user demographics and geographic location, it is important to note that there exists a significant fraction of users willing to pay a fee to use a service that is also offered for free, as long as they can enjoy some "premium features".[6]

[6]The artful reader may argue that cracked or phished Premium accounts are sometimes available. On the other hand, RapidShare can detect and block these accounts as easily as a typical user can locate and use them.

3.5.3 File popularity

Here, we examine the files shared by RapidShare clients in our client-side traces. Due to privacy concerns, we limit our analysis to the number of unique files, ignoring the actual filenames. We further examine the type of content shared using RapidShare in Section 3.8. Note that the same content object (a movie or a song, for instance) can be stored as several different files in RapidShare (aliases). The following analysis focuses on the popularity of individual files, not of the underlying content. We should first note that we only focus on file popularity as seen at our two monitoring sites. Obviously, we cannot make any statements about the popularity distribution of different files in a wider scale. Our main focus is to examine whether caching RapidShare content close to clients would make sense or not.

Figure 3.9 shows the popularity of each file as the number of clients that downloaded that file in our traces. More than 75% of the files were downloaded only once. The inner plot focuses on the distribution of the most popular files. Very few files are highly popular among the clients of each monitor; less than 0.05% of the files were downloaded more than five times, and only a handful of files were downloaded more than ten times. These results suggest that there would be little benefit to cache RapidShare files close to clients, arguing in favor of a centralized infrastructure where all servers reside at the same location. Indeed, as the next section shows, this appears to be the case with RapidShare. Also, this file popularity distribution makes RapidShare very different than traditional CDNs that rely heavily on caching popular web objects close to clients and that maintain cache hit rates of more than 90% [GDS+03, AANPF].

3.5.4 Summary

This section performed a characterization of RapidShare client behavior using client-side traces from two university networks. Our results show that more than half of the clients perform more than one file download per day. These downloads are mostly performed by non-paying users, who experience download throughput up to 2Mbps. However, a significant fraction of users (around 12%) is willing to pay a small fee to get a better service. The premium users seen in our client-side traces experience download throughput up to 14Mbps

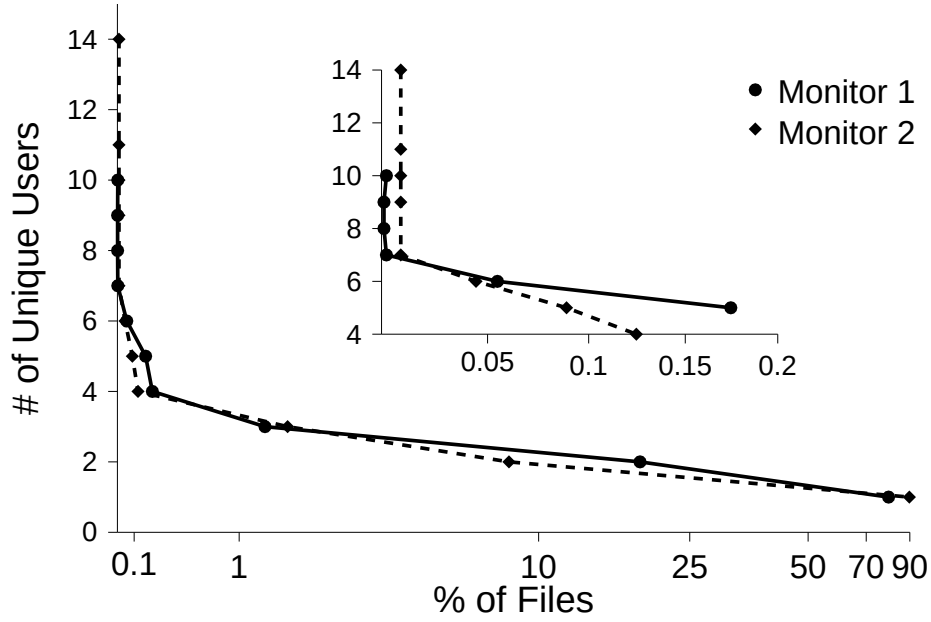


Figure 3.9: File popularity measured by the number of clients that downloaded each file.

(i.e. downloading a 200MB file in less than 2 minutes). The users' daily activity mainly involves downloading a small number of files, which often corresponds to a single media object.

In terms of traffic patterns, even though RapidShare can be viewed as just another web service, the popularity of unique files downloaded by its clients differs significantly from that of traditional web browsing. We observed only a small number of files being downloaded more than once during the whole monitoring period. This suggests that caching RapidShare content close to clients would offer little or no benefit.

3.6 Service architecture

In this section we attempt to understand the RapidShare architecture based on information from our client-side traces as well as from active measurements. In particular, we explore the number of deployed servers, their network connectivity, geographical location, load balancing and content replication strategies.

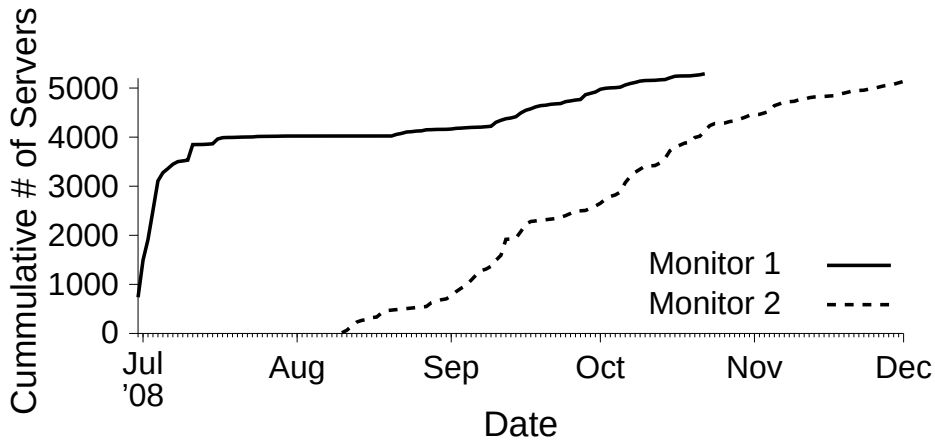


Figure 3.10: Cumulative number of RapidShare server IP addresses seen in our traces.

3.6.1 Number of servers

Figure 3.10 shows the cumulative number of RapidShare server IP addresses seen in our client-side traces. Recall that we identify such servers from the HOST header field of the HTTP requests seen at the two monitors. We observed 5,291 RapidShare server IP addresses at Monitor1 and 5,135 server IP addresses at Monitor2. The slight difference should be expected given there are more clients, and thus more sessions, at Monitor1. The increase in the number of servers in early September is interesting: RapidShare had announced that they will increase their server infrastructure at the same time period. [7]

Note that each IP address does not necessarily correspond to a distinct host. It is possible that the same host has several network interfaces, or that the same physical host is used as multiple virtual servers with distinct IP addresses. We attempted to infer, using the IP-ID method [Bel02], whether different server IP addresses generate packets with interleaved IP-ID values, but we could not find any. Of course, this test does not exclude the possibility of server virtualization, as two virtual servers on the same physical host would run different IP stacks.

3.6.2 Address blocks and upstream ISPs

At both monitors, we observed that the server IP addresses belong to 36 distinct /24 subnets (the term "subnet" refers to a /24 prefix block in this chapter). We looked up the origin-

[7] http://rapidshare.com/news.html

ISP	AS	Subnets
Level 3	3356	195.122.131/24, 195.122.149/24, 195.122.151/24, 195.122.152/24, 195.122.153/24, 212.162.63/24, 62.140.31/24, 62.67.46/24, 62.67.50/24, 62.67.57/24
GlobeInternet	6453	195.219.1/24, 80.231.128/24, 80.231.24/24, 80.231.41/24, 80.231.56/24
GBLX	3549	206.57.14/24, 208.48.186/24, 64.211.146/24, 64.214.225/24, 64.215.245/24
Fidelity	22958	207.138.168/24
INETBONE	25074	212.162.2/24
DTAG	3320	217.243.210/24, 62.153.244/24, 80.152.63/24
Cogent	174	82.129.33/24, 82.129.35/24, 82.129.36/24, 82.129.39/24
TeliaNet	1299	80.239.137/24, 80.239.151/24, 80.239.152/24, 80.239.159/24, 80.239.226/24, 80.239.236/24, 80.239.239/24

Table 3.3: Address blocks and transit providers used by RapidShare.

AS of these subnets and the results are shown in Table 3.3. The 36 subnets are allocated to 8 ISPs. This large degree of multihoming is typical for large content providers, such as RapidShare [DD08]. Multihoming can improve the reliability, performance and transit costs of a content provider. In particular, a content provider would prefer to balance its load among upstream providers so that the 95-th percentiles of its outgoing traffic through each provider remain as low as possible. There are commercial "intelligent route control" systems that perform such load balancing optimizations [GQX+04].

3.6.3 Server locations

With such a large number of servers and upstream providers, we may expect that RapidShare deploys servers in a large number of different geographical locations, similar to standard CDN practices. To explore this issue we probed the observed RapidShare server IP addresses from multiple geographical locations (landmarks) using `traceroute`. Our landmarks were

several Planetlab hosts in different countries around the globe [SPBP06, CCR+03]. Our geolocation method is simple and it is based on the minimum Round Trip Time (RTT) between each landmark and a server. Katz-Basset *et al.* [KBJK+06] showed that shortest RTT measurements using `ping` can provide geolocation results of comparable accuracy to more complex methods. An alternative and simpler method would be to use a geolocation database to query for the server's location, though some initial experiments with such a database gave as inaccurate results mapping each IP address to the location of the corresponding ISP.

Figure 3.11 shows the RTT results. The landmark locations are shown in the x-axis. Each point in this plot is the minimum RTT measurement between the corresponding landmark and the servers of a RapidShare subnet. There are 36 points for each landmark, one for each subnet. We sorted the landmarks according to the minimum measured RTT across all subnets. Landmarks with RTTs lower than 100ms are located in European countries. The lowest RTTs come from landmarks in central Europe (Netherlands, Germany and France). Landmarks in the US, Brazil or Japan give much higher RTTs for all subnets. In some cases, the subnet measurements from the same landmark are grouped in clusters; each cluster corresponds to a different routing path from the landmark to the corresponding subnets. The results of Figure 3.11 suggest that all RapidShare subnets are probably located somewhere in central Europe.

To identify the location of servers more accurately, we also examined the 2-3 last hops returned from `traceroute` towards all RapidShare IP addresses, from a single landmark. There were 48 distinct names for the penultimate hop (the last router before the destination machine): 41 of them appear to be located in Frankfurt, Germany because their names contain either the name of that city or airport/city abbreviations such as FRA and FFM. These 41 hops account for 3566 of the server IP addresses. One hop name, accounting for 743 IP addresses, contained the string *VIE*, which is the abbreviation of the Vienna airport in Austria. For the remaining penultimate hops, we had 5 IP addresses with no DNS name that account for 161 RapidShare IP addresses. Using IP-geolocation we pinned those addresses in a town in Germany close to Stuttgart (Leinfelden-Echterdingen).

In summary, our geolocation analysis suggests that RapidShare deploys all its server

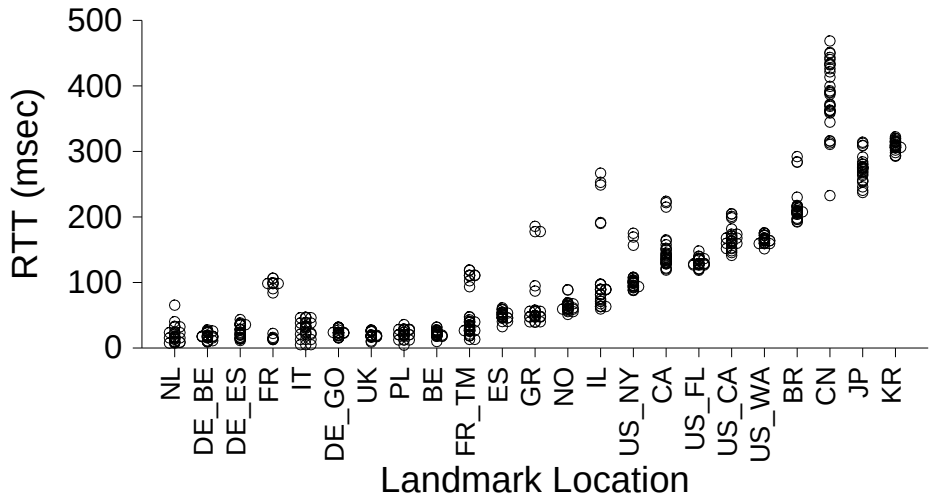


Figure 3.11: Traceroute minimum RTTs (one for each RapidShare subnet) as measured from different Planetlab landmark hosts.

infrastructure at a single location that is in (or close to) Frankfurt, Germany. We will discuss the benefits of such a centralized infrastructure later in this section.

3.6.4 Content replication and server groups

Next, we estimate how many RapidShare servers host each file. To do so, we used the MOR, a technique we have developed that uses the Tor anonymity network [DMS04] as a geographically distributed network of clients. MOR is described in detail in Chapter 5. Specifically, we first collected almost 22,000 RapidShare URLs from public indexing web sites, and then we repeatedly requested those URLs for download using 421 different Tor exit nodes around the world (thus, each Tor node appeared as a different client to the RapidShare servers).

We observed two RapidShare servers in each download request. The first is used as the "indexing server" and it returns the server name that should be used for the download. The second is the actual "download server" that sends the requested file. Interestingly, the indexing server is always the same for a given file, while each file can be served by 12 RapidShare download servers. We refer to the dozen of servers that host the same file as a "server group". Further, it appears that all servers of the same group have two properties. First, the last byte of their IP address is the same. Second, those 12 servers belong to different

subnets (of the same or different ISPs). As will be discussed later, each server group has a unique "group-ID" number that also appears in the server's name.

In summary, it appears that for each RapidShare hosted file, a unique indexing server redirects each client request to one of 12 download servers. To increase availability in the presence of ISP failures, and potentially to decrease transit fees, the download servers of the same group belong to different subnets (and often, to different ISPs).

3.6.5 Server Naming

RapidShare uses an interesting server naming scheme that allows easy identification of the upstream ISP and of the server group.

All server names start with the `rs` string. Then, the last byte from the server's IP address follows, either after reducing it by one (to have a starting point at zero) or after subtracting 1 and then adding 200, 400, or 600. The resulting number is the group-ID that the server belongs to. The next part of the name is the initials of the upstream ISP for that server. If there are several servers in the group that are connected to the same ISP, there is an additional number after the ISP initials.

For example, if a server has IP address *82.129.36.100* and its content provider is Cogent, its DNS name will be one of the following: *rs99cg.rapidshare.com*, *rs299cg.rapidshare.com*, *rs499cg. rapidshare.com*, *rs699cg.rapidshare.com*. The server's group-ID, in this example, will be 99, 299, 499, or 699. Thus, even though the last byte of the IP address can have only 256 values, the group-ID can take a much wider range of values. If there are two servers in that group connected to Cogent, the string "cg" will be followed by the number 1 or 2.

3.6.6 Server load balancing

In this section we explore the following two issues:

1. Which server group will host a newly uploaded file?

2. Which download server of that group will be used upon a download request?

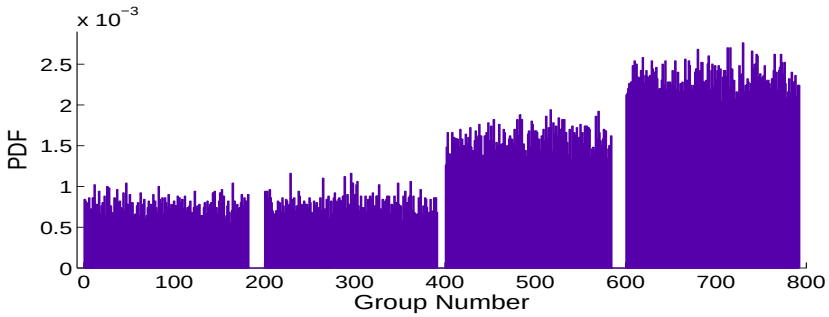


Figure 3.12: Histogram of server group-IDs assigned to new upload requests.

To answer these questions, we performed a number of active measurements. When a user attempts to upload a file, the RapidShare service first responds with a list of 12 possible server names, part of the same server group. At that point, the group-ID is already determined and it can be inferred by the servers' name. The user can then select one of those servers and perform the upload. Note that RapidShare has no information about the size or type of the uploaded file when it determines the server group.

In our first experiment we performed 50,000 back-to-back upload requests to RapidShare (without performing the actual uploads). For each upload request we logged the returned group-ID. Figure 3.12 plots a histogram of the returned server groups. Observe that there exist four different ranges of group-IDs with different frequencies: 0-200, 200-400, 400-600 and 600-800. The first two have similar frequencies. It appears that the last two ranges correspond to large service expansions that RapidShare performed in Sep'08 (400-600) and in Mar'09 (600-800). Thus, more recently deployed servers have larger group-IDs, as one would probably expect. It is interesting that servers with larger group-IDs get a higher likelihood of upload assignments. Our interpretation of these results is that more upload requests are assigned to recently deployed servers, which should also have more available capacity, attempting to gradually balance the hosting load.

Our second experiment focuses on the download server selection process. Since each file is hosted by a group of 12 servers, how is the server that will handle a new download request selected? We performed one thousand back-to-back download requests for the same file and

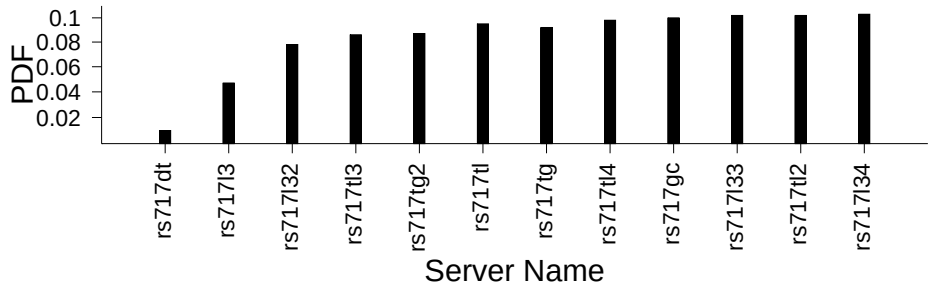


Figure 3.13: Histogram of servers from the same group assigned to new download request.

logged the returned download server. Figure 3.13 shows the histogram of the download servers for the corresponding group (group-ID: 717). Note that 10 out of the 12 servers have a very similar likelihood of serving the download requests. The two remaining servers are selected with lower probability. Server `rs717l3` is also used as the indexing server for this group, and it probably receives a lower download load due to its double role. Server `rs717dt` has an even lower download load, but we do not understand why.

In summary, it appears that RapidShare uses simple load balancing rules, allocating more upload requests to more recently deployed servers, and assigning more download requests to servers that do *not* also act as the indexing server of the group. On the other hand, it seems that RapidShare does not try to control which ISP will be used for each download request (if they were doing so, the distribution in Figure 3.13 among the 10 download servers would not be uniform).

3.6.7 Discussion and comparison with CDNs

In this section, we explored a typical OCH service architecture Unfortunately, the service does not disclose such information and so we cannot determine the accuracy or correctness of our conclusions. Our results suggest that the architecture uses few thousands of servers (and the number keeps increasing) that are multihomed to several ISPs (some of them tier-1 transit providers). All servers are in the same location, probably close to Frankfurt in Germany. The servers are partitioned in groups of 12 servers each, while the servers of the same group belong to different subnets (and are often connected to different ISPs). Each file is hosted by one server group. It appears that upload requests are assigned to groups in a manner that

considers the "deployment age" of each group. The actual upload server within the returned group, and the corresponding ISP, can be selected by the user. In terms of download requests, a user is directed to a specific download server from an indexing server, which is a member of the group that hosts that file. The assignment of downloads appears to be uniform across the download servers of the group.

The OCH architecture differs significantly from traditional web Content Distribution Networks (CDNs), such as Akamai [aka]. CDNs maintain mirror servers in many geographic locations, as they attempt to minimize the RTT between the user and the mirror server that will serve that user. It is not uncommon for a large CDN to be present at hundreds of geographical locations around the world. This difference with OCH is reasonable: a traditional CDN aims to minimize web transaction delays and it is optimized for short TCP flows. An OCH service such as RapidShare, on the other hand, focuses on very large transfers that are less sensitive to delay. The centralized architecture of OCH is probably less expensive, easier to maintain, and it makes the migration of files between servers faster and less costly. Another difference is that the main value that CDNs offer to their customers is improved web performance. For OCH services, performance is less important compared to content availability and the ability to share files inexpensively. Finally, the business models of CDNs and OCH services are very different. The former get their revenues from large content producers, while the latter get their revenues from individuals that subscribe to Premium accounts.

3.7 Comparing Rapidshare and BitTorrent

File sharing has traditionally used the p2p paradigm. p2p file sharing, mostly using BitTorrent, is the dominant source of traffic in the Internet today. In this section we explore whether the dominance of the p2p paradigm for file sharing can be challenged by the emerging OCH paradigm. Do OCH services provide significant benefits, in terms of performance or content availability, over p2p applications for file sharing?

To answer this question, we compare RapidShare, the leading OCH service, with BitTorrent, the leading p2p system, along the dimensions of download throughput and content

availability. We think that these two factors are the most important from the typical user's perspective. Cost is another important factor, of course, when a user considers subscribing for a Premium RapidShare account. It should be noted that the following comparisons can only consider the current deployment of these two services; obviously, we cannot know whether the performance of RapidShare would deteriorate if that service was handling the same number of users as BitTorrent.

3.7.1 Download throughput

To compare performance, we manually downloaded 38 (non copyrighted) files from both services. The list of downloaded objects was randomly constructed from a number of indexing web sites, and it included a variety of files. The sizes of the selected objects range from 1.6MB to 2.85GB. Each content object was present in both services with approximately the same file size and quality. [8] We made sure that none of our downloads were illegal, requesting files that are not copyrighted. In the case of BitTorrent, we selected those torrents that had the larger population of seeders, to get the best download throughput. All downloads were done from the same client at FORTH. The client was connected to the public Internet through a 1Gbps access link.

For RapidShare, we used three different types of users. The first role is a **Premium user**. The second role is a **Free user**. Free users can download only one file at a time, and they are throttled to a throughput between 0.2Mbps and 2Mbps. Further, Free users have to wait for about 15 minutes between two successive downloads. In our measurements, we included these long wait times in the total download latency that a Free user experiences when she downloads an object that consists of several files. The third role is again a Free user that is able to change her IP address, through a new DHCP request, so that she can avoid the mandatory waiting period between consecutive downloads. We refer to such users as **Free-cheating**. [9]

Figure 3.14 shows the distribution of average throughput for each of the three user roles.

[8] In some cases, we could find the exact same object uploaded in both BitTorrent and RapidShare.

[9] It appears that **Free-cheating** is a popular behavior among OCH users. Many content indexing sites offer instructions on how to avoid the long waiting time that Free users have to experience. Of course, it is possible to do so when the user can acquire different IP addresses quickly and easily.

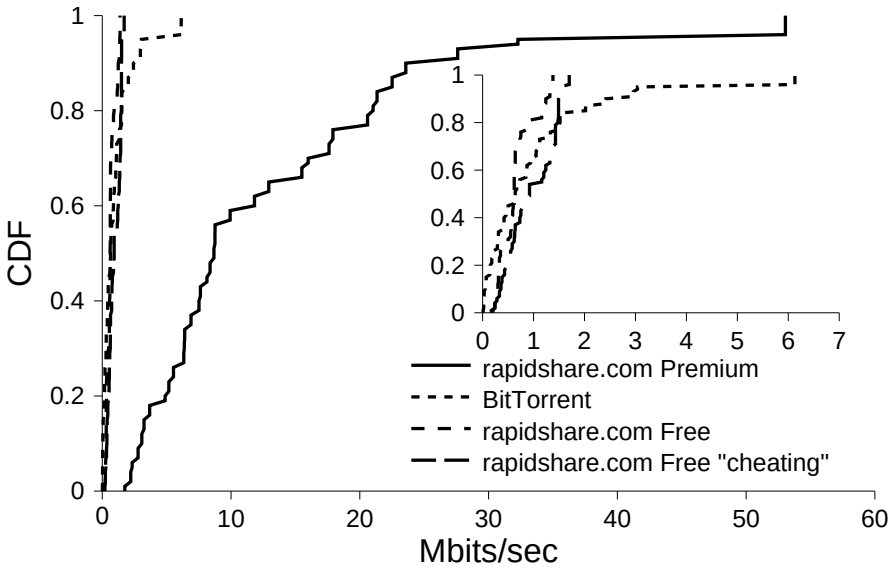


Figure 3.14: Comparison of download throughput between three RapidShare user roles and BitTorrent.

Note that **Premium** RapidShare users enjoy a very high throughput compared to BitTorrent and the two other user roles. Indeed, more than 50% of the Premium RapidShare downloads got a throughput 8Mbps or more - an order of magnitude higher than BitTorrent downloads. Of course, RapidShare Premium downloads are a paying service and many users would prefer to not pay for file sharing, given that p2p services are also free. The inner plot of Figure 3.14 suggests that the median download session experiences higher throughput with the Free-cheating RapidShare service than with BitTorrent. Indeed, 50% of the Free-cheating RapidShare downloads receive more than 920kbps, while the Free RapidShare and BitTorrent median download throughput is about 600kbps. A small percentage of BitTorrent downloads however receive much higher rates than Free-cheating RapidShare downloads. The fastest 10% of BitTorrent downloads receive more than 2.4Mbps.

3.7.2 Content availability

We next focus on the issue of content availability in the two services. To do so, we collected well-known lists of movies and songs, such as the Internet Movie Database [10], and searched for them in both RapidShare and BitTorrent. To verify that a specific object exists on RapidShare we first searched for it using Google, including the term "rapidshare" as a keyword together with the title of the object, and then examined the most relevant RapidShare link. We rejected search results that would not contain the full name of the object we search for. Then, we requested each collected URL from RapidShare to examine if it is still available for download (but without downloading the file).

To see whether a file exists on BitTorrent, we searched for it on `piratebay.org`, the most popular torrent search site on the time of the experiment that hosts more than 600,000 torrents. The file was considered available if there was at least one seeder for it. [11]

Table 3.4 presents the related results. The second column shows the number of files contained in the object list. The third column is the number and percentage of objects that we were able to find on RapidShare using Google searches. The fourth column presents the number and percentage of objects found on BitTorrent. We see that in all cases, RapidShare has higher content availability than BitTorrent. For instance, in the "All Time USA Box Office" hits, 98.5% of the objects were found in RapidShare and 96.5% were found in BitTorrent. The difference between RapidShare and BitTorrent becomes more important for the less popular content. In the "Bottom 100 movies" list, RapidShare hosts 90% of those movies, while BitTorrent hosts only 50% of them. Users that are interested in this type of content seam to prefer RapidShare instead of BitTorrent. We speculate that users may prefer the simplicity of sharing files through a web page with URLs to those files in contrast to using a tracker and a specific file-sharing application.

[10] www.imdb.com

[11] This means that 100% of the file was available by at least one peer.

List Name	Number of objects	Found on RapidShare	Found on piratebay.org
All Time Non-USA Box Office	385	377 (97.9%)	373 (96.8%)
Bottom 100 movies as voted by users	100	90 (90%)	53 (53%)
All Time USA Box Office	408	402 (98.5%)	394 (96.5%)
Top 250 movies as voted by users	250	245 (98%)	240 (96%)
All Time Worldwide Box Office	346	338 (97.7%)	336 (97.1%)
Top United States DVD Rentals for week ending 16 November 2008	50	50 (100%)	50 (100%)
Amazon Best German films of all time	25	21 (84%)	19 (76%)

Table 3.4: Availability of movie objects in RapidShare and piratebay.org (BitTorrent).

3.7.3 Summary

Our results suggest that RapidShare users enjoy a better file sharing experience than BitTorrent users. Indeed, Premium RapidShare users enjoy an order of magnitude higher download throughput than BitTorrent users. Even free RapidShare users have the benefit of more available content on RapidShare than on BitTorrent. Considering the actions a user has to follow to download an object, we believe that the process is equally simple in both services. In the case of BitTorrent, the user has to first search for the torrent file and then instrument her BitTorrent client to download that object. Similarly, a RapidShare user needs to first search for the object and download it through her browser. When an object spans several files, and thus several URLs need to be downloaded, there are simple "download managers" for web browsers that can reduce the entire procedure to just a handful of mouse-clicks. Uploading files is much easier in RapidShare because the user only uploads a file in whole or in pieces. In BitTorrent one has to first find (or deploy) a hosting tracker, create the torrent file, and maintain an always-on host that will be serving the file until the swarm gets big enough. Considering these differences, file sharing user communities may have good reasons to use RapidShare instead of BitTorrent.

Name	# of Indexed Objects	RapidShare Hosted Objects	# of Stale Files	# of Uploaders
egydown.com	972	787	134 (17%)	N/A
rapidmega.info	942	893	116 (13%)	9
rslinks.org	12124	11841	64 (0.5%)	21
rapidshareindex.com	54327	36522	7052 (19.3%)	18

Table 3.5: Description of four OCH content indexing sites.

3.8 Content indexing sites

As previously mentioned, OCH services do not offer search or indexing capabilities. To fill this gap, a large number of content indexing web sites offer such capabilities to OCH users. These indexing sites form several communities of users, ranging from general interest to very specific interest content. In this section we explore some of these sites to better understand how users search for objects in RapidShare, to examine the population of users that post download URLs on indexing sites, and to characterize the publicly visible RapidShare content in terms of type and copyright constraints.

3.8.1 Content uploaders

First, let us examine the community of RapidShare users that post download links on OCH indexing sites. To do so, we crawled four such sites shown in Table 3.5. The second column presents the number of OCH URLs provided by each indexing site. For example, rapidshareindex.org provides URLs to 54,327 objects hosted by OCH services; 36,522 of those are hosted by RapidShare. It is interesting that these 54,327 objects were uploaded by only 18 users! A very small number of uploaders is what we also observe in other indexing sites.

Figure 3.15 shows the percentage of posts by each user of three indexing sites. Again, we observe that a small number of users post almost all download URLs. For instance, in rslinks.org, only 5 users have posted more than 90% of the URLs, while in rapidshareindex.com a single user has posted (almost) all URLs.

As described in Section 3.2, files hosted by RapidShare become stale/invalid when the uploader requests to remove them, or when the uploader is a Free user and the file has either been downloaded 10 times or it has stayed inactive for a period of 90 days. Table 3.5 also

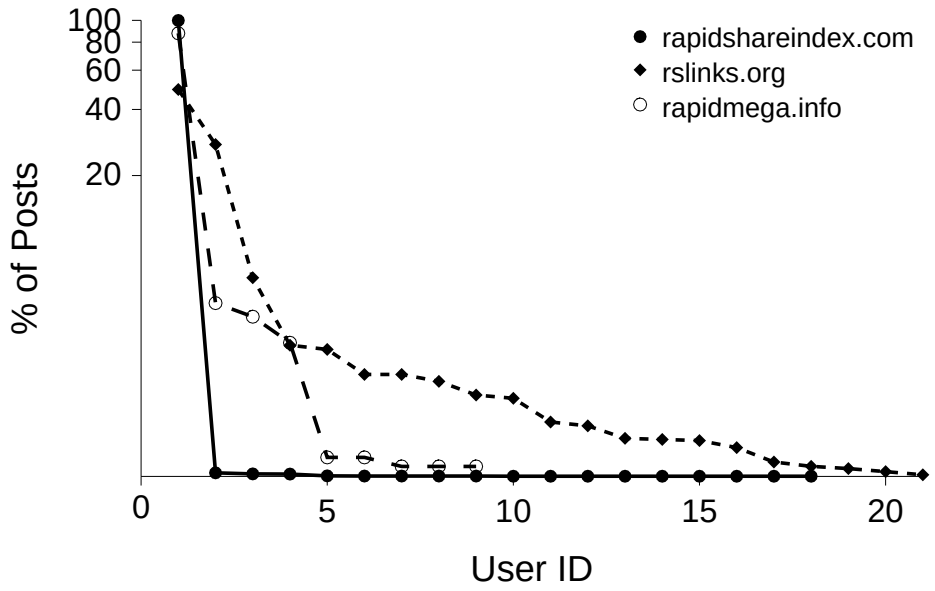


Figure 3.15: Percentage of posts per uploader at three content indexing sites.

shows the number and percentage of stale objects. We retrieved the state of each object by requesting it for download from RapidShare, and then checking for a valid download response. Stale files count for at most 19.3% of all files, and in one case for as little as 0.5%. This is in sharp contrast with file availability in p2p systems. For example, BitTorrent files have an average lifespan of only 9 days [GCX+05].

An interesting possibility is that the percentage of stale files is more community-specific, rather than service-specific. RapidShare could also have a large fraction of stale files, especially for files uploaded by Free users. However, if a user community cares about a file, they would not let it become stale by successively uploading "fresh" versions. Based on the indexing sites we examined, we can say that this seems to be the case for RapidShare user communities.

3.8.2 Characterization of publicly visible content

Currently, RapidShare allows Free users to download files up to 200 MB. Thus, most of the large objects that are publicly shared are partitioned in files of that size (or smaller), and the user has to download a number of URLs. For instance, a full DVD of 2.4GB would be available through 12 URLs. In order to identify the type of objects available in RapidShare we crawled the previous four indexing sites and counted the number of download URLs per

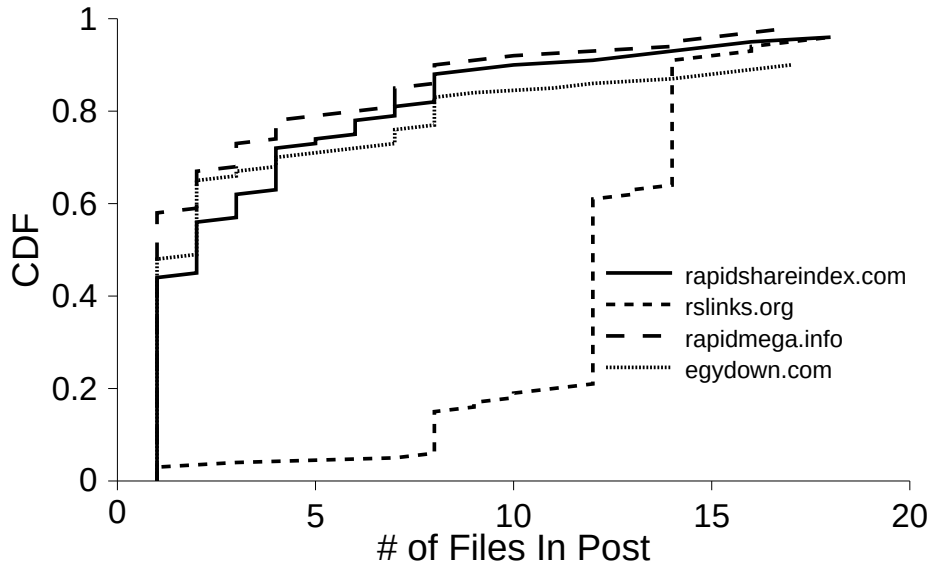


Figure 3.16: Percentage of download URLs per indexed object.

Name	Games	Video	Apps	Books	Images	Audio
egydown.com	11	19	65	1	4	0
rapidmega.info	0	45	1	1	0	53
rslinks.org	0	100	0	0	0	0
rapidshareindex.com	0	21	74	0	3	2

Table 3.6: Classification of 100 objects from each content indexing web site.

object.

Figure 3.16 shows the distribution of the number of URLs per object for each indexing site. At most 60% of the objects consist of a single URL. In the case of `rslinks.org`, this percentage drops to as low as 3%. A closer look to that site reveals that it is mainly used to share large video files (movies and TV series count for 80% of its total objects) and application/game CDs (20%).

To get a more clear picture of the type of objects that are publicly shared through OCH services, we manually examined the latest 100 objects seen at each of the previous four indexing sites. The results are summarized in Table 3.6. We see that video files, applications and audio collections are the largest categories - as expected. Again, `rslinks.org` is only used for movies.

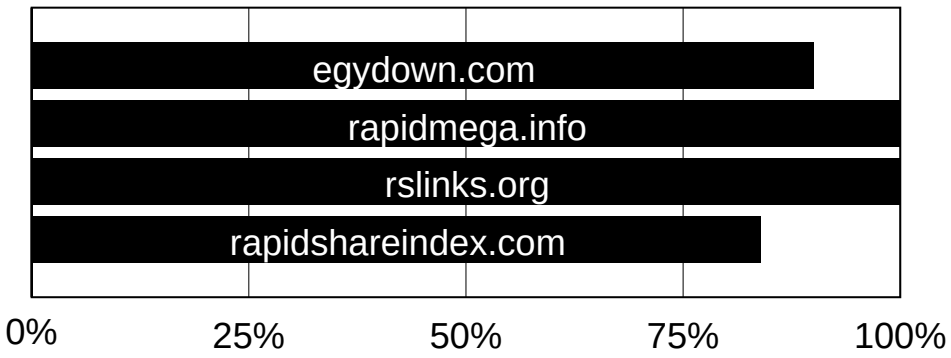


Figure 3.17: Percentage of copyrighted material in a sample of 100 objects from each indexing site.

3.8.3 Copyrighted content

Internet file sharing, through p2p networks or through OCH services, is constantly accused of copyright infringement. To get a rough estimate of the fraction of copyrighted content indexed in public OCH indexing sites, we manually examined the 100 most recent objects listed on each content indexing web site. We classify an object as copyrighted if is a commercial movie, song, book or application, based on the official web page of the corresponding content. Figure 3.17 shows the percentage of objects which appear to be copyrighted content. More than 84% of the 100 most recent objects, and in some sites 100% of the objects, appear to be copyrighted.

We note that OCH services, including RapidShare, are *not* responsible for any illegal file sharing and copyright infringements that take place using their infrastructure. RapidShare, in particular, aggress to host a file only if the uploading user declares that the content of that file is not copyrighted. [12] Further, RapidShare does *not* provide indexing and search capabilities for the content they store. Consequently, any copyright infringements that take place using their infrastructure is solely a responsibility of the users that upload and download such content.

[12] As is clearly stated in the "Terms of Use" aggreement `http://rapidshare.com/agb.html`.

3.9 Conclusions

Using a combination of passive monitoring and active probing we provide the first, to our knowledge, detailed study of OCH services, and of RapidShare in particular. Our results show that OCH services have reached a level of popularity at which *they produce a significant share of the daily web traffic: a share that in some cases exceeds the traffic generated by YouTube and similar on-line video services.*

Most users download more than one file during their daily sessions. Interestingly enough, even over several months of observation, we saw very little locality of reference in the objects downloaded through OCH services, probably reflecting the diverse interests of file sharing users. These locality patterns imply that there will be no significant benefits by caching content near to users, which indicates that a centralized architecture would be appropriate for OCH services. Indeed, our experiments revealed that RapidShare employs a centralized, heavily multi-homed server infrastructure that is located at a single geographical location. RapidShare servers are grouped together into groups of 12, with each server in the group hosting the same set of files. Each server of the same group belongs to a different IP subnet and it is often connected to a different ISP. With frequent infrastructure expansions, RapidShare tries to keep its storage capacity evenly distributed among all server groups, by favoring recently added servers for new file uploads.

Comparing RapidShare with BitTorrent, we found that the former provides better performance than the popular p2p network. Our experiments suggest that *although Free RapidShare users experience similar throughput with BitTorrent users, Premium RapidShare users experience an order of magnitude higher throughput.* Furthermore, the amount of content shared over RapidShare is larger than the content shared with BitTorrent. Furthermore, the availability of content in RapidShare appears to persist for longer time periods than in BitTorrent. With a significant fraction of users willing to pay a small fee to enjoy premium service, OCH services appear likely to compete with BitTorrent as the leading file-sharing platform.

We also examined OCH content indexing sites, which are an essential component for file sharing using OCH services. We found that in OCH services, much like in p2p file sharing

systems, *a very small number of users upload most files, which are often copyrighted content*, favoring audio albums, video movies, and applications.

Chapter 4

Short URL hosting services

In this chapter, we provide the first characterization on the usage of short URLs. Specifically, our goal is to examine the content short URLs point to, how they are published, their popularity and activity over time, as well as their potential impact on the performance of the web.

4.1 Overview

URL shortening has evolved into one of the main practices for the easy dissemination and sharing of URLs. URL shortening services provide their users with a smaller equivalent of any provided long URL, and redirect subsequent visitors to the intended source. Although the first notable URL shortening service, namely tinyURL [tin], dates back to 2002, today, users can choose from a a wide selection of such services. [1] The recent popularity of shortening services is a result of their extensive usage in Online Social Networks (OSNs). Services, like Twitter, impose an upper limit on the length of posted messages, and thus URL shortening is typical for the propagation of content. While short URL accesses represent a small fraction of the *"web hits"* a site receives, they are rapidly increasing by as much as 10% per month according to Alexa [ale].

Despite this rapid growth, there is, to the best of our knowledge, no other large-scale study in the literature that sheds light onto the characteristics and usage patterns of short URLs.

[1] http://www.prlog.org/10879994-just-how-many-url-shorteners-are-there-anyway.html

We feel that understanding their usage has become important for several reasons, including:

i. Short URLs are widely used in specialized communities and services such as Twitter, as well as in several Online Social Networks and Instant Messaging (IM) systems. A study of URL shortening services will provide insight into the interests of such communities as well as a better understanding of their characteristics compared to the broader web browsing community.

ii. Some URL shortening services, such as bit.ly have grown so much in popularity, that they now account for as much as one percent of the total web population per day [ale]. If this trend continues, URL shortening services will become a critical part of a web page's access procedure posing challenging questions regarding its performance, scalability, and reliability. We believe that answering these questions and defining the proper architectures for URL shortening services without understanding their access patterns is not feasible.

To understand the nature and impact of URL shortening services, we perform the first large-scale crawl of URL shortening services and analyze the use of short URLs across different applications. Our study is based on traces of short URLs as seen from two different perspectives: i) collected through a large-scale crawl of URL shortening services, and ii) collected by crawling Twitter messages. The first trace provides insights for a general characterization on the usage of short URLs. The second trace moves our focus onto how certain communities use shortening services. The highlights of our work can be summarized as follows:

1. We study the applications that use short URLs and show that most accesses to short URLs come from IM Systems, email clients and OSN media/applications, suggesting a "word of mouth" URL distribution. This distribution implies that short URLs appear mostly in ephemeral media, with profound effects on their popularity, lifetime, and access patterns.

2. We show that the short URL click distribution can be closely approximated by a log-normal curve, verifying the rule that a small number of URLs have a very large number

of accesses, while the majority of short URLs has very limited accesses.

3. We study the access frequency of short URLs and observe that a large percentage of short URLs are not ephemeral. 50% of short URLs live for more than three months. Further, we observe high burstiness in the access of short URLs over time. Short URLs become popular extremely fast suggesting a "twitter effect", which may create significant traffic surges and may pose interesting design challenges for web sites.

4. We show that the most popular web sites (as seen by the number of short URLs accesses towards them) changes slowly over time, while having a strong component of web sites which remains stable throughout the examined period. Our experiments also suggest that the web sites which are popular in the short URL community differ profoundly from the sites which are popular among the broader web community.

5. We examine the performance implications of the use of short URLs. We find that in more than 90% of the cases, the resulting short URL reduce the amount of bytes needed for the URL by 95%. This result suggests that URL shortening services are extremely effective in space gaining. On the other hand, we observe that the imposed redirection of URL shortening services increases the web page access times by an additional 54% relative overhead. This result should be taken into consideration for the design of future URL shortening services.

4.2 URL Shortening Services

The idea behind URL shortening services is to assist in the easy sharing of URLs by providing a short equivalent. For example, if the user submits `http://www.this.is.a.long.url.com/indeed.html` to bit.ly, the service will return the following short URL to the user: `http://bit.ly/dv82ka`. The user can then publish the short URL on any webpage, blog, forum or OSN, exactly as she would use the original URL. Any future access to `http://bit.ly/dv82ka` will be redirected by bit.ly to the original URL through an "HTTP 301 Moved Permanently" response.

URL shortening services have existed at least as early as 2001 [mak]; tinyURL [tin] is

probably the first such, well-known, service. The rapid adoption of OSNs, and their imposed character limit for status updates, tweets and comments, has led to an increased demand for short URLs. As a consequence, dozens of such services exist today, although only a handful of them, such as bit.ly, ow.ly and tinyURL, capture the lion's share of the market. Aside from the aforementioned services, short URLs are also useful in more traditional systems which either discourage the use of very long words, such as IMs and SMSes, or do not handle long URLs very well, such as some email clients.

Besides providing a short URL for each long one, some of these services provide statistics about the accesses of these URLs. For example, bit.ly provides information about the number of hits each short URL has received (total and daily), the referrer sites the hits came from and the visitors' countries. For each unique long URL that it has shortened, bit.ly provides a unique global hash, along with an information page which provides the overall statistics for the URL. If a registered user creates another short URL for the same long URL, the service will create a different hash that will be given to the user so as to share it as she likes. The information page for this custom hash will contain statistics solely for the hits received by the creator's URL. Nonetheless, overall statistics will still be kept by the global URL's information page. Registered users can create as many custom short URLs as they like for the same long URL.

4.3 Data Collection

This section introduces our data collection process and gives a description of the collected data. Overall, we study short URLs from two different perspectives: i) By looking at two shortening services, namely bit.ly and ow.ly, and ii) by examining short URLs and their usage within OSNs, and, in particular Twitter.

4.3.1 Collection Methodology

We use two approaches to collect short URLs: i) *Crawling*, in which we search Twitter to find tweets which contain URLs and ii) *Brute-Force*, in which we crawl two URL shortening

services, that is bit.ly and ow.ly, by creating hashes of different sizes and examining which of them already exist.

As mentioned in the previous section, bit.ly maintains an information page for each created short URL. This page provides detailed analysis regarding the amount of hits a short URL received, its HTTP referrers and the geographical locations of its visitors. The daily amount of hits since the creation of the short URL is also recorded. Information regarding the number of hits from each referrer and country is provided as well. For each bit.ly short URL in our traces we also collect the accompanied information pages. Information pages for short URLs created by registered users also contain a reference to the global short URL for this long URL. For the sake of completeness, our analysis includes the information provided by the global hash. Unfortunately, ow.ly does not provide any such information.

Twitter Crawling: Using the first method, we search for HTTP URLs that were posted on Twitter. Using the Twitter search functionality [twib], we collect tweets that contain *HTTP* URLs. Twitter imposes rate limiting in the number of search requests per hour from a given IP address [twia]. To respect this policy we limit our crawler to one search request every 5 minutes. Every search request retrieves up to 1500 results (tweets), going no more than 7 days (max) back in time. During our collection period we managed to collect more than 20 million tweets containing HTTP URLs. 87% of the collected HTTP URLs were short URLs. Among the HTTP URLs collected from Twitter, 50% were bit.ly URLs. The second most popular shortening service was tl.gd with 4%, while tinyURL corresponded to 3.5% and ow.ly amounted to 1.5% of the overall URLs. Hence, part of our analysis focuses on bit.ly URLs.

Brute-Force: Using the second method, we exhaustively search the available keyspace for ow.ly and bit.ly hashes. While the Twitter crawling approach returns links recently "gossiped" in a social network, this approach acts as an alternate source of collection, providing hashes irrespective of their published medium and recency.

In the bit.ly case, we searched the entire keyspace [0-9a-zA-Z] for hashes of up to 3 characters in length. Currently, the shortening service returns 6-character hashes, indicating a significant exhaustion of shorter combinations. In the case of ow.ly, the system does not

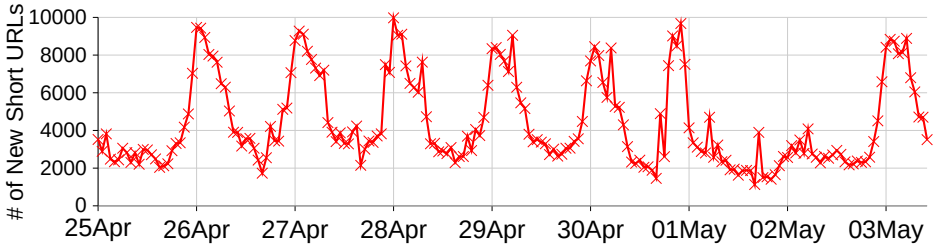


Figure 4.1: Number of ow.ly short URLs created as a function of time.

disseminate random hashes of the user's long URL but serially iterates over the available short URL space; thus, if the same long URL is submitted multiple times, it will result in multiple different hashes. Considering this deterministic registration mechanism, we collected the full set of short URLs created for a period of 9 days. During that time, we monitored the evolution of the keyspace by creating a new short URL of our own every hour and measuring the distance from the one we had created the previous hour. Using this heuristic, we were able to determine which and how many short URLs were created during that timeframe with a granularity of one hour. Figure 4.1 shows the number of ow.ly URLs registered as a function of time. As expected, we observe a clear diurnal and weekly cycle, with about 70,000 new short URLs created each day.

Having collected sets of bit.ly short URLs with the aforementioned methods, we proceed with the gathering and analysis of the metadata provided by the shortening service. Initially, we access the corresponding information page and record the resulting long URL, the total number of hits it has received, the name of the user that created it and the global short URL, offering aggregated data. We go on to collect the daily history of hit events for the entirety of the short URL's lifespan. Furthermore we fetch the number of hits per referrer and country. Finally, we follow the global short URL and download the aggregated versions of the metadata as well.

4.3.2 Collected Data

The previously discussed collection process resulted in four datasets:

trace name	service	number of URLs	accesses	first URL access	last URL access
twitter	bit.ly	887,395	101,739,341	2008-07-08	2010-04-29
twitter2	bit.ly	7,401,026	2,202,442,600	2008-06-27	2010-09-25
owly	ow.ly	674,239	not available	2010-04-26	2010-05-03
bitly	bit.ly	171,044	15,096,722	2008-07-07	2010-05-06

Table 4.1: Summary of data collected

- *twitter*: The trace contains 887,395 unique bit.ly short URLs posted on Twitter between the 22nd of April and the 3rd of May 2010. For each short URL, all the accompanied metadata are also collected.

- *twitter2*: The trace contains over 7M unique bit.ly short URLs posted on Twitter between the 6th of May and the 2nd of August 2010. In this trace we limit our metadata gathering to only the total and daily accesses for each short URL.

- *owly*: This trace contains 674,239 ow.ly short URLs created between the 26th of April and the 3rd of May 2010. As described in the brute-force methodology, this constitutes the entire population of ow.ly short URLs created in that period.

- *bitly*: Contains 171,044 unique bit.ly short URLs collected by exhaustively searching the available key space for hash sizes of 1 to 3 characters. All the accompanied metadata for each short URL are also collected.

Table 4.1 summarizes the data collected.

4.3.3 Representativeness

Before proceeding with the analysis of the collected data, we first examine the representativeness of these traces. To provide an estimation on the ratio of tweets that contain bit.ly URLs, we retrieved the total number of tweets, for a specific time window, using the public timeline feature of the Twitter API. For the same time window, we also collected the total number of tweets containing bit.ly, through the live search feature of Twitter. We examined both quantities for 144 10-minute windows, for the total period of 1 day. On average, we observed that 4.9% of all posted tweets contained bit.ly short URLs. With our relaxed

Rank	*twitter*		*bitly*	
	Site	% of Accesses	Site	% of Accesses
1	eMail,IM,apps,phone,direct	59.32	email,IM,apps,phone,direct	72.72
2	twitter.com	23.49	twitter.com	11.77
3	partners.bit.ly	3.02	www.cholotube.com	2.16
4	www.facebook.com	2.17	www.facebook.com	1.72
5	healthinsuranceexchange.info	1.57	partners.bit.ly	1.63

Table 4.2: The 5 most prolific Referrers of short URLs.

crawling methodology we managed to retrieve about 7% of all new tweets containing one or more bit.ly short URLs. To estimate the benefit of a more aggressive crawling methodology, we used a second crawler, deployed only for the limited time period of a single day, issuing a search request every thirty seconds. The aggressive approach was able to harvest almost four times more tweets than the moderated one.

As discussed in Section 4.4, our findings remain the same when comparing statistics across the two crawling rates. The only observable difference is that, as expected, a more aggressive rate results in the collection of a larger number of less popular short URLs, i.e., short URLs that received one or two hits. Taking into consideration the ethical aspects of web crawling and considering that our tweet sampling ratio was large enough to allow the extraction of valid characteristics and behaviors, we followed the relaxed collection rate for the results presented throughout the chapter.

4.4 The web of Short URLs

We begin our analysis with a general characterization of short URLs. Over the following sections, we identify where short URLs originate from, the type of content they point to, and analyze their popularity patterns.

4.4.1 Where do short URLs come from?

Despite the fact that short URLs are typically seen within OSN services, URL shortening services have already existed for a number of years. Thus, a natural question to ask is whether there are particular communities of users or applications where the usage of short

URLs is dominant.

To this end, we study the "referrers" of each short URL, information that is provided by bit.ly for each short URL. Table 4.2 lists the top-5 most popular referrers for the URLs in traces *twitter* and *bitly*. We see that in both cases the vast majority of users (that is, 60% and 72% respectively) arrive at bit.ly from non-web applications; these include Instant Messaging and email clients, mobile applications like Twitterific and BlackBerry mail, Twitter desktop applications and directly (by pasting/typing the URL in a browser). For those users that do access short URLs through web applications, we observe that they mostly come from Twitter, and various other social-networking-related sites. This suggests that bit.ly (and possibly other URL shortening services) are most popular in social networking applications/communities.

The distribution of referrers in Table 4.2 reveals an entirely new browsing model for short URLs users. According to our findings, short URLs do not frequently appear in traditional web pages but are distributed via Instant Messaging (email,IM,phone) and social network channels (twitter.com, facebook.com), suggesting a "word of mouth" type of propagation. This has significant impact on the browsing habits and patterns of short URL users as we show in the following sections.

4.4.2 Where do short URLs point to?

Having observed that short URLs mostly originate in non-browser type of applications, we now aim at understanding the type of web pages that are popular through bit.ly links. To achieve this, we manually classified the content of the 100 most accessed domains in the *twitter* trace. Similarly, we classified the links of the *owly* trace, which was obtained via the Brute-Force method and presents a perhaps more general view of the content served through short URLs. In the case of ow.ly, the number of accesses per short URL is not available so we selected the most popular domains based on the number of shortened URLs under each domain.

Table 4.3 presents the top categories for each case. One may notice that news and informative content come first. This observation corroborates the finding of Kwak et al. [KLPM10], which suggested that Twitter acts more as a information-relaying network rather than as a

twitter		*owly*	
Category	% Sites	Category	% Sites
news (inc. portals)	25	news (inc. portals)	51
info / edu	18	various	17
various	13	info / edu	10
entertainment	10	social networking	5
personal	9	media sharing	5
twitter-related	9	shorten urls	4
commercial	6	commercial	4
media sharing	4	twitter-related	2
social networking	4	sharing articles	1

Table 4.3: Most popular types of content.

social networking site. However, while this study suggests that trending topics are related to news by as much as 85%, the fraction of news related short URLs is significantly lower in our case (25% and 51% for the two traces). A surprising finding is that 4 of the most accessed URLs in the *owly* trace were shortening services. Such cases reflect short URLs packed inside other short URLs to avoid exposure of the long URLs from tools that unwrap the first level of redirection. Spammers use such techniques to avoid detection, as mentioned by Grier et al. in [GTPZ10]. Manualy examining a number of these URLs confirmed this suspission with a large number of short URLs pointing to spam content. We plan further inverstication of this phenomenon as future work.

4.4.3 Location

We now examine the geographic coverage of short URL usage, i.e., whether short URL users follow the distribution of Internet/web users or whether short URLs are a niche application of some particular countries. Table 4.4 shows the distribution of the country of origin of short URL accesses in the *twitter* and *bitly* traces. Most of these accesses come from the United States, Japan, and Great Britain. Interestingly enough we do not see any accesses from China and India, which are ranked in the top-5 countries with the largest number of Internet users [int]. Our conjecture is that applications which use short URLs are probably not popular or widespread in the above countries, suggesting that the penetration of short URL use is significantly different from the Internet/web one.

Rank	*twitter*		*bitly*	
	Site	% of Accesses	Site	% of Accesses
1	US	42.12	US	54.15
2	JP	12.20	GB	5.59
3	None	8.95	None	4.83
4	GB	5.96	CA	4.14
5	CA	4.58	PE	3.48
6	BR	4.41	JP	2.80
7	DE	3.68	BR	2.21
8	FR	1.77	DE	2.14
9	NL	1.25	AU	1.57
10	AU	1.24	IN	1.51

Table 4.4: The 10 Countries with the largest number of clicks.

4.4.4 Popularity

As discussed in Section 4.4.2, short URLs primarily refer to news and other information related content. In this section, we examine the particular domains visited through short URLs, and their popularity over time. First, however, we examine the popularity distribution of individual URLs. Popularity is measured by examining the number of hits a URL received.

URL Popularity: Large systems that provide content to users typically exhibit a power-law behavior [BCF+98, RFI02] with respect to the offered content (e.g., [CKR+07]). That is, a small fraction of the content is very popular, while most of it is considered uninteresting, characterized by moderated access rates. Figure 4.2 (top) depicts the popularity distribution of the short URLs in the *twitter* and *twitter2* trace, and the corresponding Cumulative Distribution Function (CDF) –bottom. As is the case with other content provider services, the distribution has a heavy tail.

Figure 4.2 also plots the popularity distribution and corresponding CDF for the short URLs collected through the aggressive harvesting, pressented in Section 4.3.3. As we observe the sampling rate we employ on the Twitter crawling method does not bias our findings. The only observable difference is that, as expected, more aggressive sampling result's in the collection of a large number of less popular short URLs, i.e., short URLs that received one or two hits.

Since our trace might be populated with recently created URLs, the distribution may be

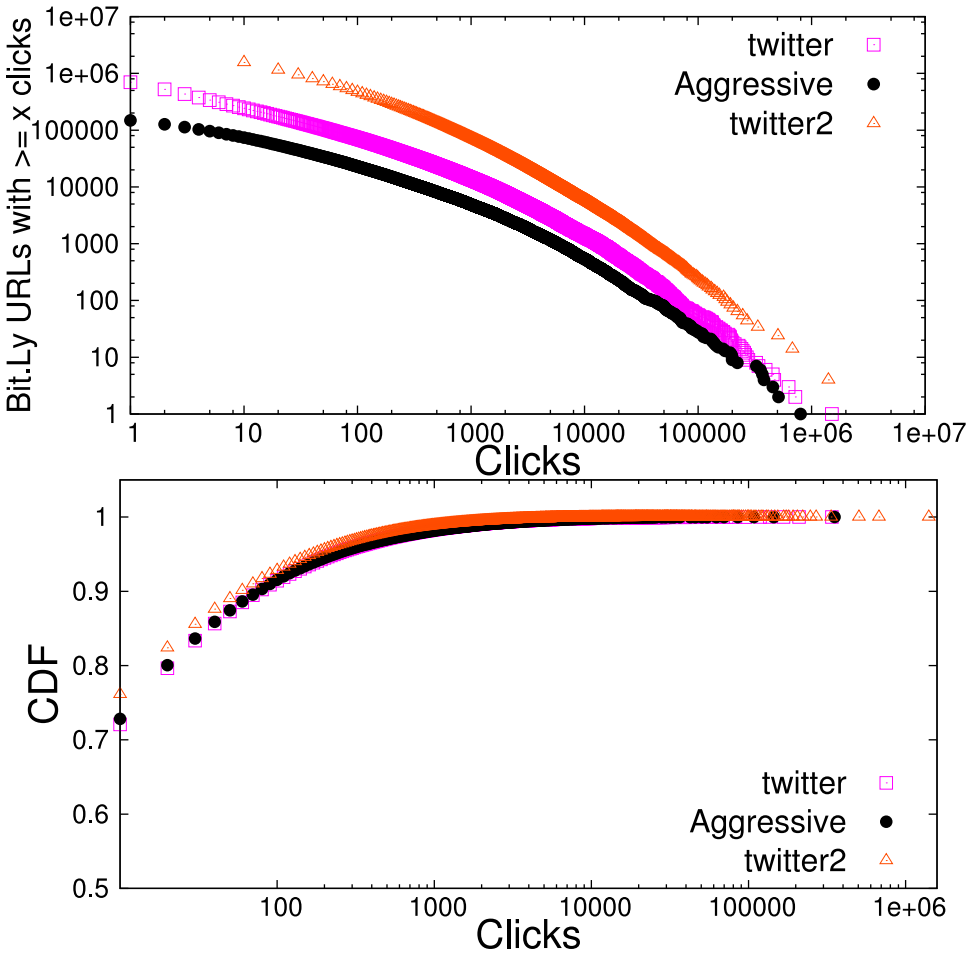


Figure 4.2: Popularity of bit.ly URLs.

biased. To examine this hypothesis, we eliminate all short URLs whose creation was during the last week of our trace collection period. Further, we split short URLs into *active* and *inactive*. As "inactive", we consider short URLs for which no hit was observed during the last week of our trace. To define the inactivity threshold for our study we experimented with several different values. Figure 4.3 shows the popularity distribution for threshold values from 7 to 56 days. Using threshold values larger than 7 days does not affect the popularity distribution. [2]

Figure 4.4 separately examines the distribution of the active and inactive short URLs for the *twitter2* trace. Both curves appear similar to the original distribution. Further, a 90-10 rule seems to apply to the distribution. That is, we see that 10% of the short URLs are

[2]Similar results were observed when examining the lifetime curve for different inactivity thresholds.

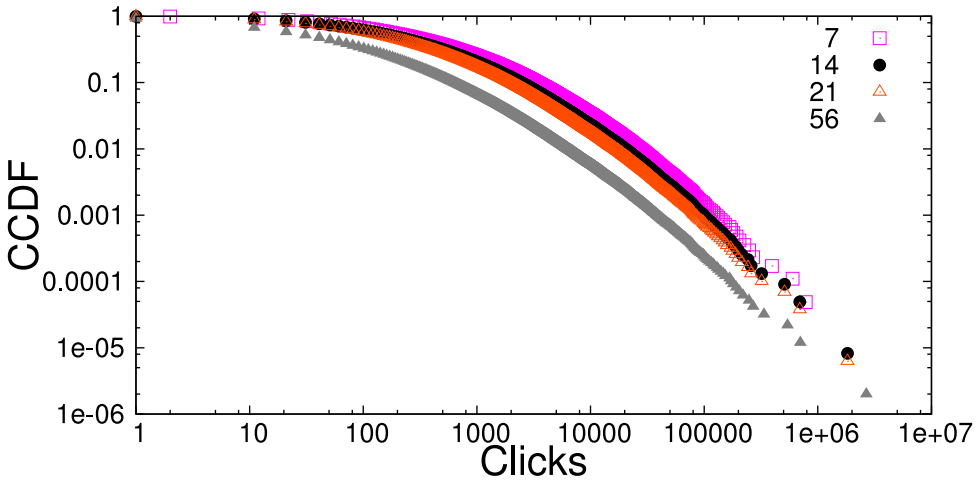


Figure 4.3: Popularity of bit.ly URLs using different activity thresholds.

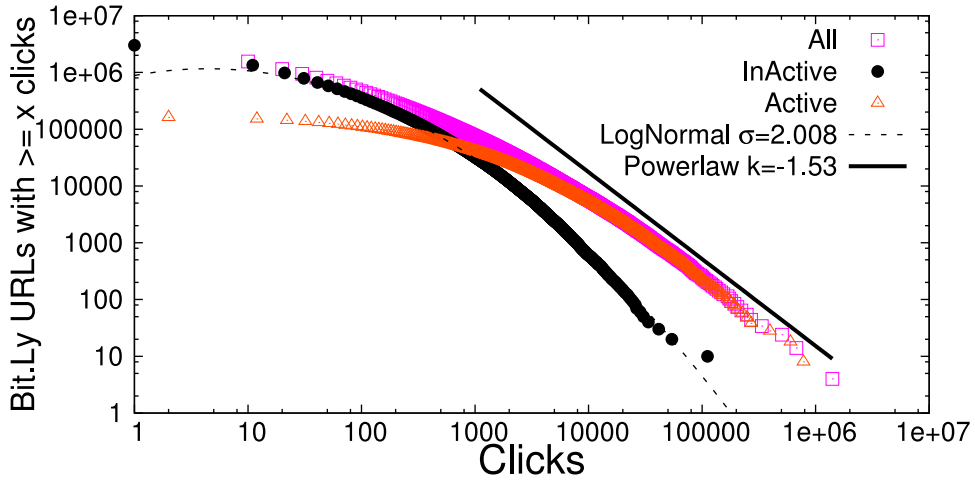


Figure 4.4: Popularity distributions for Active and In-Active bit.ly URLs from *twitter2* trace.

responsible for about 90% of the total hits seen in our trace.

Content Popularity: So far we have analyzed the overall popularity of individual short URLs, and examined its distribution. We now proceed to study *which web sites people access using short URLs*. Using the daily access information from the *twitter* and *bitly* traces, we try to answer questions such as:

i. Which are the most popular web sites accessed through short URLs?

ii. Are these sites similar to the ones found in the "traditional" web?

iii. Does the set of these popular web sites change over time, and if so, how?

Rank	*twitter*				*bitly*			
	Site	% of Accesses	Alexa Rank	NetCraft Rank	Site	% of Accesses	Alexa Rank	NetCraft Rank
1	www.youtube.com	10.42	3	3	winebizradio.com	15.2	2693058	N/A
2	mashable.com	2.14	315	1175	www.youtube.com	10.51	3	3
3	www.facebook.com	1.91	2	2	livesexplus.com	3.98	15250029	N/A
4	www.47news.jp	1.51	3376	14605	mashable.com	2.28	315	1175
5	pollpigeon.com	1.24	57842	153550	inws.wrh.noaa.gov	2.27	1169	N/A
6	www.omg-facts.com	1.1	N/A	150669	www.alideas.com	2.26	7536010	N/A
7	twibbon.com	0.76	21271	55376	about:blank	1.87	N/A	N/A
8	itunes.apple.com	0.75	52	673	googleblog.blogspot.com	1.63	2251	2223
9	www.newtoyinc.com	0.72	167768	988477	addons.mozilla.org	1.56	247	197099
10	www.guardian.co.uk	0.65	273	231	www.google.com	1.53	1	1

Table 4.5: The 10 most popular web sites as seen through the real user accesses of the bit.ly URLs in traces *twitter* and *bitly*.

Table 4.5 lists the 10 most popular web sites: that is, the sites which received the highest numbers of hits through the short URLs in the two traces. Surprisingly, besides familiar sites, such as Youtube and Facebook, we observe others that are less known or popular according to well known ranking services such as Alexa and Netcraft; for example, `pollpigeon.com` (a service for very short opinion polls), `mashable.com` (a social media news site), `twibbon.com` (a Twitter campaign support site), etc. Note that the list does not significantly change when using the data collected through aggressive crawling (Section 4.3.3), nor the larger Twitter trace (*twitter2*). This further supports that our selected sampling gives a good representation of the overall statistics collected through Twitter.

As we have observed previously, short URLs are mostly found in social networking or interaction environments and, thus, their popularity reflects the interests of the particular communities. For example, taking short polls is very common in social networking sites. Thus, such URLs rank very high in accesses through short URLs, even though they may not rank high in a more general web browsing environment. Overall, our findings indicate that while the community which browses the web through short URLs shares some interests with the broader web browsing community, it also presents a distinctive focus on web sites of special interest.

In addition to identifying the popular web sites, we are also interested in understanding whether these web sites significantly change over time. To this end, we calculated the 100

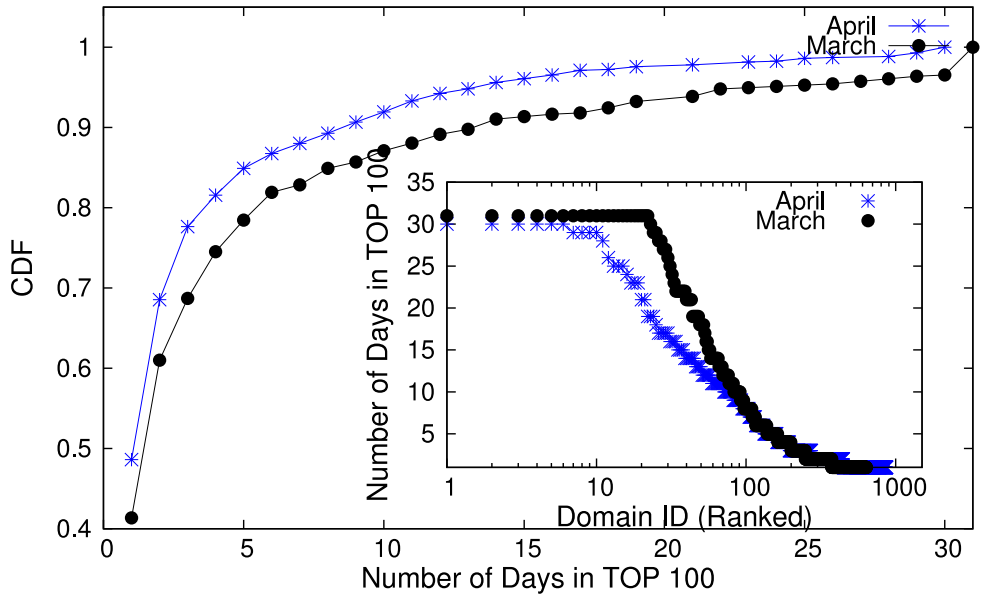


Figure 4.5: Number of days a domain name is in TOP-100 during March and April 2010.

most popular web sites per day for the entire months of March and April 2010. 868 and 636 different sites where present in the daily top-100 respectively. Figure 4.5 displays the number of days a site appears in the top-100 each month. The Figure shows that there are about 6 sites which appear every single day of April 2010 in the top-100 (22 sites for March 2010). These compose a kernel of popular sites which does not seem to change over time, and has captured the attention and interest of bit.ly users. Additionally, we see that there are about 400 sites which appear once or twice in the top-100, enjoying short bursts of popularity. The results for the top 10 most popular web sites per day show similar behavior in a smaller scale. We further examine this burstiness effect in detail in Section 4.5.

4.5 Evolution and lifetime

The analysis throughout the previous section highlights the fact that short URLs differ from traditional URLs in many ways. Being published through social networking applications (Section 4.4.1), they have inherent idiosyncrasies that affect their observed activity over time. Indeed, the liveness of a short URL depends on factors such as the visitor's activity and her screen real estate. Since news feeds in social network environments typically display

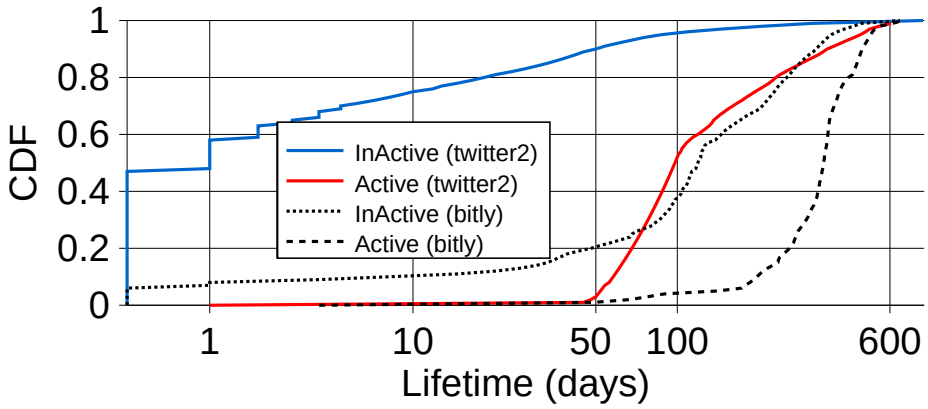


Figure 4.6: Lifetime analysis of short URL in traces *twitter2* and *bitly*.

recent activity and are frequently updated, once a short URL disappears from the visitor's screen, it has almost no chances of getting clicked. Furthermore, short URLs are not directly "searchable" and are far from easy to remember, therefore users rarely access them explicitly.

In this section, we analyze how active a short URL is, by examining its hit rate over time. Specifically, we ask the following questions:

i. Are short URLs ephemeral or do they survive for long periods of time?

ii. How is the hit rate of a short URL spread across its lifetime?

We consider such queries pertinent to the cacheability of short URLs that provide implications for the design of shortening services (e.g., URL recycling).

4.5.1 Life Span of short URLs

To examine the life span of short URLs we focus our attention on the *twitter2* and *bitly* traces. Both traces refer to the same shortening service which provides the daily hit rate per short URL. We define the life span, or *lifetime*, of a URL as the number of days between its last and first observed hit.

Figure 4.6 displays the lifetime CDF of the two traces. The figure further splits URLs into *active* and *inactive*, as these are defined in section 4.4.4. Recall that, as "inactive", we consider all short URLs for which no hit was observed during the last week of our trace. This

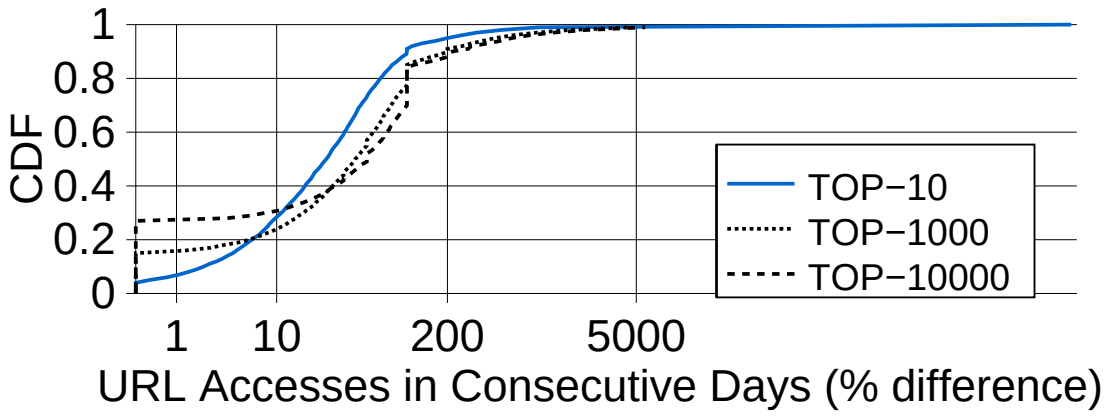


Figure 4.7: Cumulative Distribution Function for the daily click differences for the TOP-10/1000/10000 short URLs.

split provides a feel of how the lifetime distribution depends on the activity of the URL, and will also be clarified in the following section when we examine the temporal characteristics of the URL hit rate.

One out of two short URLs are not ephemeral! While one might expect that short URLs are mostly ephemeral URLs, i.e. lasting for a few days, the aforementioned figure shows that 50% of the active short URLs for the *twitter2* and *bitly* traces have a lifespan of 98 and 124 days respectively. On the other hand, inactive URLs have a shorter lifespan as expected, with 51% only lasting for a day for the *twitter2* trace. Still a significant fraction of short URLs (more than 15%) last at least one month.

4.5.2 Temporal evolution

Having observed that a significant fraction of URLs survives for numerous days, we will now turn our focus on how hits are spread throughout a URL's lifetime. For the remainder of this section, we will use the *twitter2* trace, unless otherwise specified.

Looking at the evolution of the number of hits per day per URL as a function of time for several high volume URLs we observe several distinct patterns. Some show sudden increases or spikes while others have a significant decrease in hit rate. However, in all cases the bursty nature of access patterns was evident.

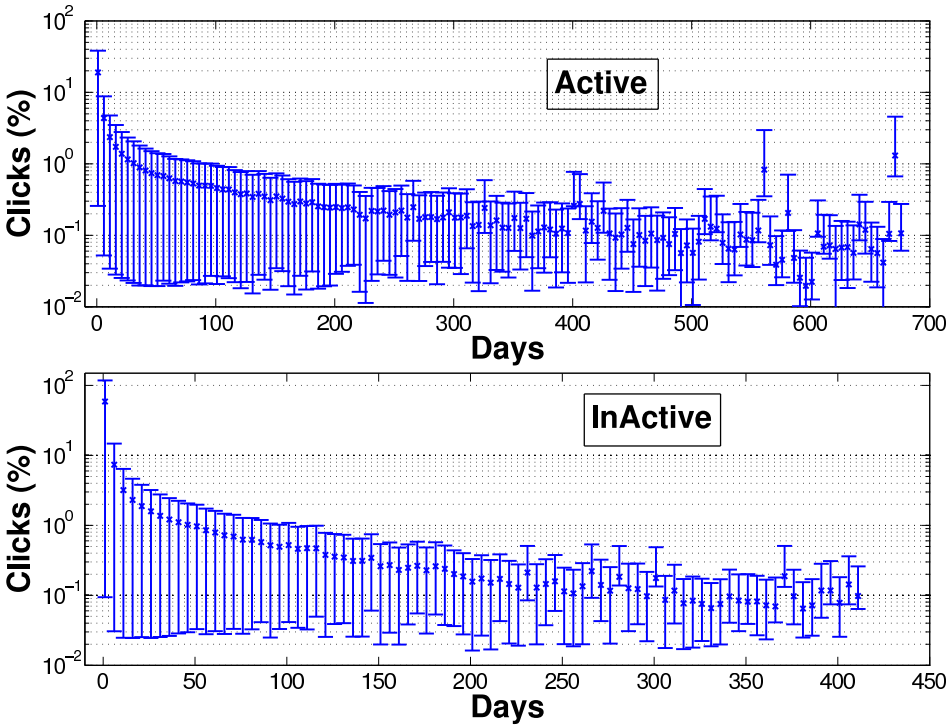


Figure 4.8: Mean and confidence intervals for the fraction of daily hits over the total clicks versus the lifetime of the URL.

We attempt to characterize this burstiness in a more generic fashion across several URLs, by measuring the daily change in the number of hits for each short URL for the top-10, top-1000 and top-10000 short URLs (see Figure 4.7). We observe that the median value is around 24% for the top-10 URLs and around 40% and 50% for the top-1000 and top-10000 URLs. In other words, the number of accesses for a typical short URL varies by as much as 40% from one day to the next. Moreover, for 10% of the days, this change is at least 100% for the top-10 and around 200% for the top-1000 and top-10000 URLs. Overall, we notice that as less popular URLs are included, that is as we move from the top-10 to the top-1000 and top-10000, we observe increasingly larger daily changes. This reflects the existence of URLs that only enjoy a few days of high popularity, and are then "forgotten".

1 day of fame. We further examine the evolution of hit rate across the lifetime of the short URLs in Figure 4.8, where we examine the mean, and confidence intervals of the fraction of a short URL's total hit rate over its lifetime, across all short URLs (with 0 denoting the

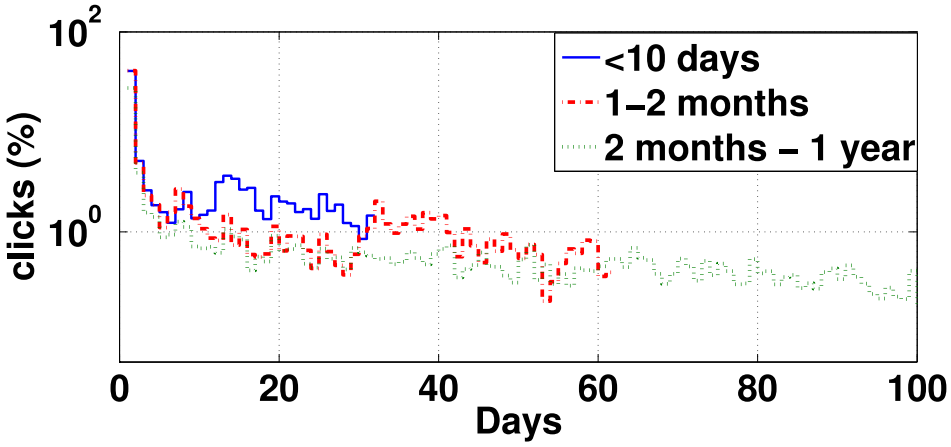


Figure 4.9: Fraction of hits per day conditioned on different lifetimes.

creation day of the short URL). The figure depicts both active (top) and inactive (bottom) short URLs which show two distinctive patterns. For the inactive URLs, we observe that on the average 60% of hits are observed during their first day. As a short URL ages, its hit rate drops sharply and then stays roughly constant as the hit ratio converges to 0. In fact, this observation holds irrespective of the lifetime of the short URL (see Figure 4.9). In contrast, while this first-day effect is also evident for active short URLs albeit with at a smaller fraction (at roughly 18%), we also observe a significant hit rate for recent days. This reflects popular short URLs that still enjoy a significant hit rate.

As previously mentioned, Figure 4.9 shows no obvious dependence of the daily hit rate with a short URL's lifetime for inactive short URLs. We examine this relationship in more detail by looking at the total number of hits as a function of the short URL's lifetime (Figure 4.10, median hit rate). The figure is in accordance with our previous observation for the inactive short URLs (top), namely that no obvious relationship exists. On the contrary, active short URLs (bottom) appear to exhibit a linear relationship in log-log scale with the lifetime of the URL.

Summarizing our discussion in this section, contrary to our expectations, we observe one out of two short URLs are not ephemeral. More than 50% of the active short URLs tend to live for more than three months. Moreover, a large number of short URLs enjoy occasional hits that may skew their lifetime. This implies that design mechanisms for shortening services

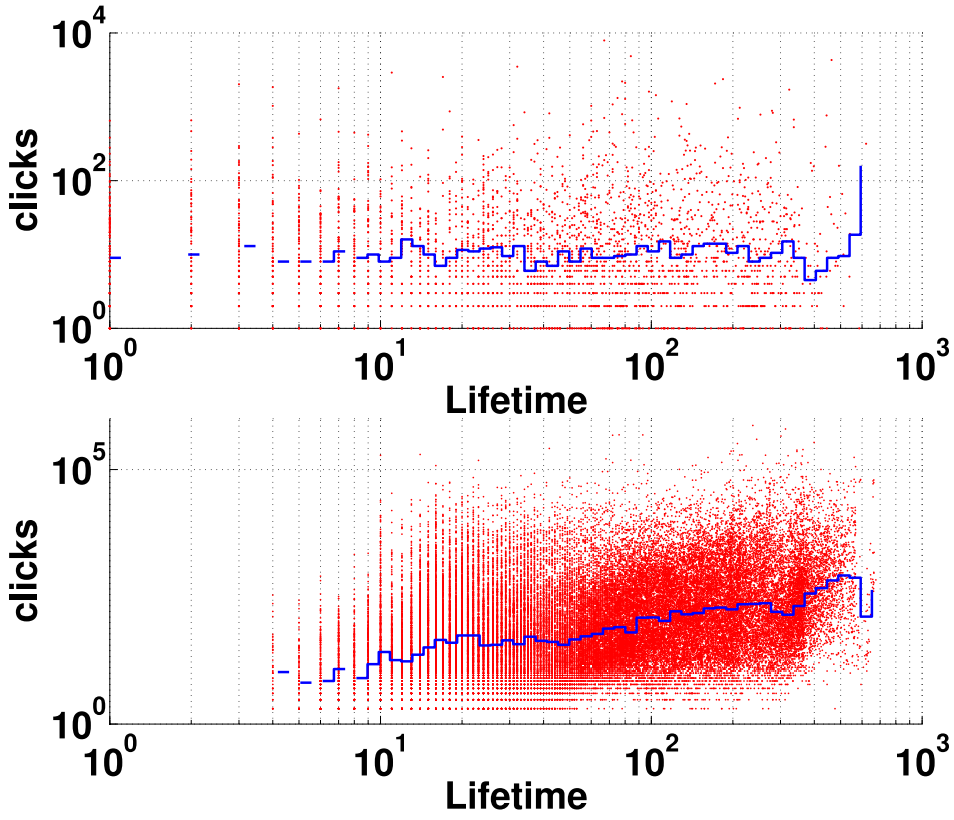


Figure 4.10: Lifetime of a short URL vs. number of hits.

should not expect a short lifespan of short URLs that is in the order of days. In addition, most short URLs enjoy a high hit rate relative to their total hits during their first day of creation, with the fraction of hits significantly dropping after.

4.6 Publishers

In this section, we focus our interest on the publishers of short URLs, i.e., users who include short URLs in Twitter messages. Twitter provides a unique opportunity for users to easily increase the popularity of their published content in a social network, which may not be possible with some of the other short URL sources. Figure 4.11 confirms this hypothesis by plotting the popularity of short URLs that received at least one hit from a Twitter user versus the popularity of all other short URLs. The *Twitter effect* is obvious: short URLs referred from Twitter enjoy significantly higher popularity compared to short URLs not experiencing

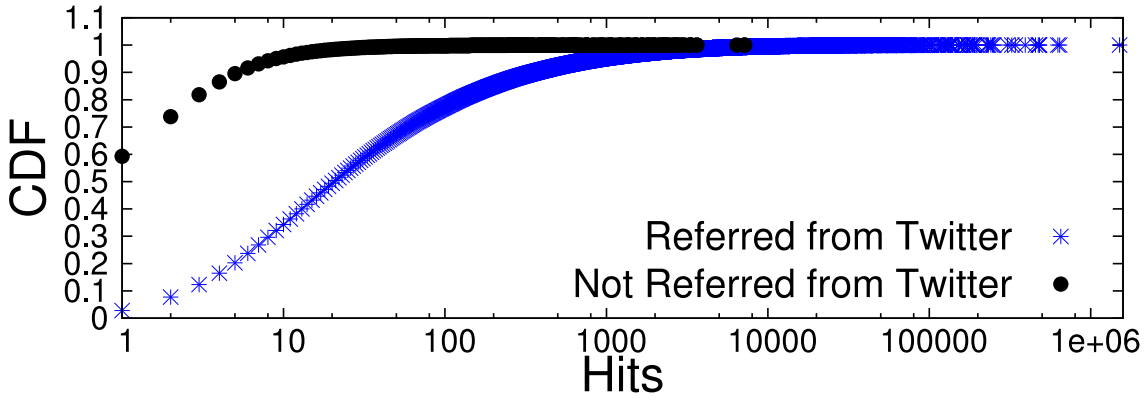


Figure 4.11: *The Twitter effect.* Difference in popularity for Twitter referred short URLs vs. non-Twitter referred ones.

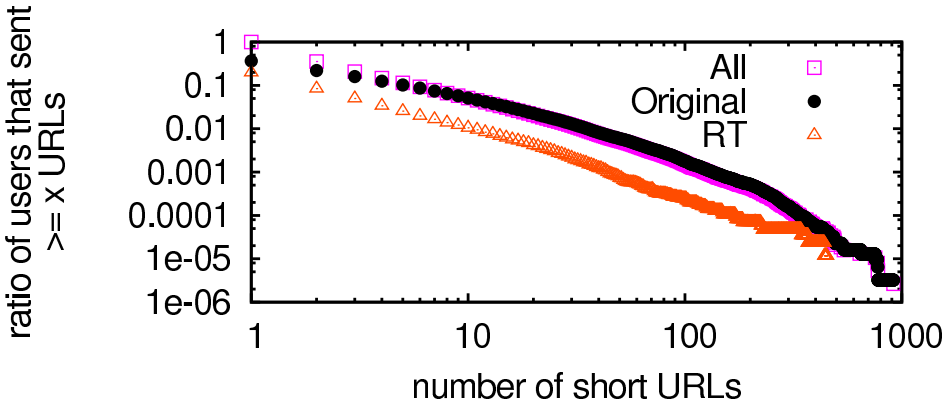


Figure 4.12: **CCDF of posted short URLs per Twitter user.** The distribution is heavy-tailed with a small percentage of users posting a large number of short URLs

this type of "word-of-mouth" propagation. Thus, examination of the publish rate and the popularity of published tweets relates to the propagation of *User Generated Content (UGC)* within social networks (e.g., [CKR+07, CMG09]), although the content reflected by the short URL in this case might not have been generated by its publisher. Note that Twitter messages may reflect original messages or "retweets", i.e., messages that are re-postings of an original message.

Our driving questions are:

i. What does the distribution of published URLs per user look like? Are there any automated users which publish disproportionately large numbers of short URLs?

ii. What is the activity of a typical user? This question relates to the publish rate of new URLs over time. Furthermore, do most users publish original URLs or retweet existing ones?

iii. Does a higher publish rate per user imply a higher hit rate for the URLs published? This is pertinent to the propagation of a user's published URL and the population this URL may reach.

Figure 4.12 plots the Complementary CDF (CCDF) of posted short URLs per Twitter user. Most users published a handful of tweets with short URLs (the median is equal to 1 short URL). Overall, 90% of the users generated 5 or less such tweets each, and 65% of the users generated only one tweet containing a short URL. On the other hand, we see that some users generated hundreds of such tweets. For example, the most prolific user generated just under one thousand such tweets. Interestingly, the majority of tweets with short URLs are original Twitter messages and not retweets (RT).

Publishing about a thousand tweets in a week is an impressive number of published messages. For this reason, we now focus on the most prolific publishers in order to understand their behavior. We subsequently inspected the profiles of the top 12 publishers. Each tweet carries a label indicating the way it was posted, i.e., via the web site, the official API or a third-party application. From these top publishers, 10 uploaded their messages via twitterfeed [twic] and the other two via TweetDeck [twe] and the API respectively. Twitterfeed is an application designed specifically for automatically relaying the contents of an RSS feed via tweets. Furthermore, we visually identified bursty message patterns in all profiles with tweets coming in batches of two or three, every few minutes. All the above clearly indicate a semi-automated behavior.

To examine the users' daily publish rate of short URLs, Figure 4.13 displays the corresponding CDF. We observe that the median rate is 1 short URL per day, while 98% of the users publish no more than 5 short URLs per day. For prolific publishers we also observe a high number of short URL in a daily basis, also explained by the several automated applications used by Twitter users.

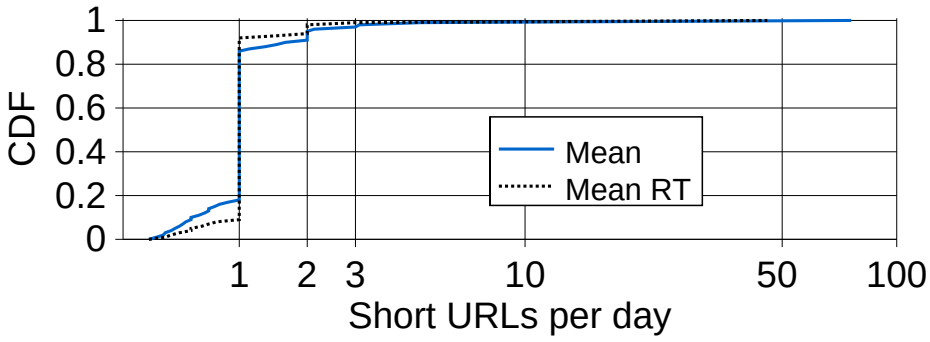


Figure 4.13: Number of posted short URLs per day per user.

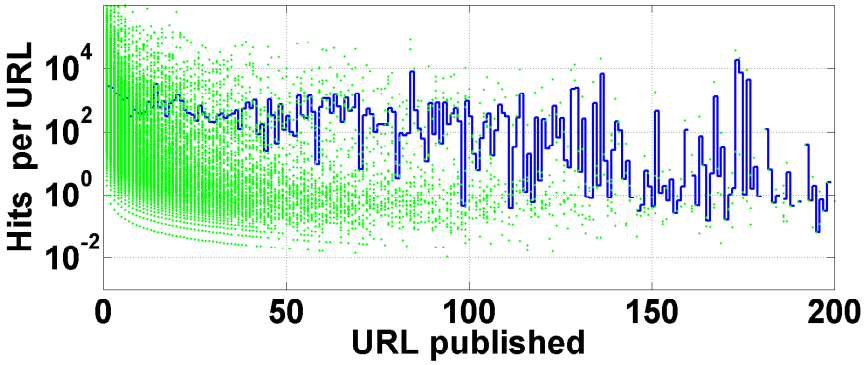


Figure 4.14: Expected hits as a function of the URLs published per user.

Intuitively, a users' publish rate should correlate with the total number of hits observed for his published URLs. However, the nature of this relationship is not evident, and depends on whether a users' followers indeed click on the posted short URL. For example, spammers or advertisers may not observe as many hits for subsequent published URLs. We examine this relationship in Figure 4.14, which displays the expected hits per URL as a function of the published URLs across users. We see that as the number of URLs published by a poster increases, the expected hit rate drops. This may imply either spamming-type behavior for heavy publishers, or that only a few short URLs from each publisher enjoy high hit rates compared to the rest of the user's published short URLs.

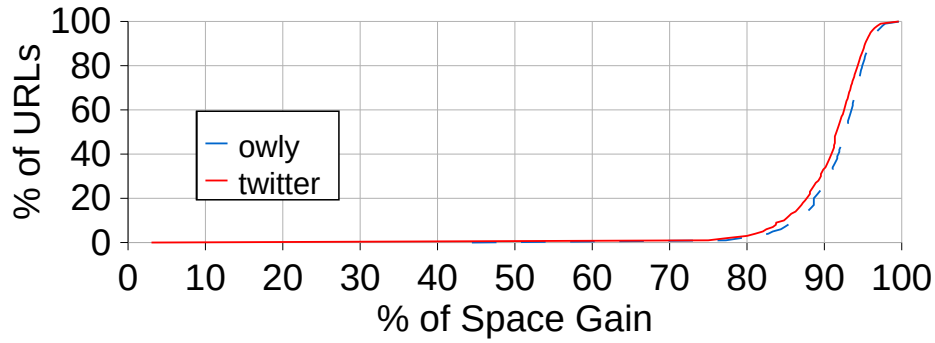


Figure 4.15: Reduction in URL size achieved by URL shortener services.

4.7 Short URLs and Web performance

Having studied the access patterns of short URLs, we now turn our attention to understanding potential performance implications of their use. We consider two such cases, namely:

i. To what extent do short URLs offer space reduction compared to long ones?

ii. short URLs introduce an extra step of indirection in the process of accessing web content. Hence, we attempt to quantify the performance penalty of this extra step. For example, could it turn out to be a major performance bottleneck?

4.7.1 Space Reduction

In this section we explore the amount of space saved through URL shortening services. As gain, we define the relative ratio of the URLs' length before and after the shortening service. Figure 4.15 displays this gain for the short URLs in traces *twitter* and *owly*. For roughly 50% of the URLs, we observe a 91% reduction in size, or about a factor of 10. Furthermore, for 90% of the URLs, the short version takes up to 95% less space than the long one - a factor of 20 improvement. Therefore, we see that URL shortening services are *quite* effective at reducing URL size and can provide significant benefit in environments where space is at a premium. A real-world approximation of the space saved by short URLs is the case of Twitter, where users place short URLs in their messages. While each tweet is limited to 140 characters, we assume that users, who would not be able to fit a long URL in their message, would either

create a second tweet or not tweet at all. In our *twitter* trace, we replaced the bit.ly URLs in all tweets with their equivalent long versions and found that only 31% remained under the character limit.

4.7.2 Latency

Although URL shortening services offer a substantial space benefit over long URLs, they nonetheless impose an additional indirection in the user's web request. This may result in an increased web page access time, user-perceived latency and an overall degradation of performance. In this section, we quantify the latency such URL shortening services add to the overall web experience by exploring whether this imposes a significant overhead in web access times.

To estimate the overhead added by URL shortening services, we periodically accessed the 10 most popular short URLs in each of four such services, namely bit.ly, ow.ly, tinyURL.com and fb.me, as seen in the *twitter2* trace. Each short URL was accessed every 5 minutes for a time frame of 30 days. For each access we logged the total time of the web page transfer and the time needed for the redirection imposed by the URL shortening service. Figure 4.16 shows the extra cost incurred due to the redirection. Three of the services are closer together, exhibiting a median value of this overhead in the order of 0.37 seconds, while, in any case, none of them lies lower than 0.29 seconds. The fourth service, fb.me, a Facebook.com shortening service, appears to have a much smaller median value, in the order of 0.16 seconds and a lower bound very close to that. However it exhibits a bimodal behavior in terms of latency with 75% of redirections imposing no more than 0.17 seconds delay and 25% slowing down the user's requests by more than 0.33 seconds. Furthermore, the distance between the fastest and slowest 5% of accesses is 0.272 seconds. ow.ly shows a similar bimodal behavior with 66% of redirections imposing less than 0.33 seconds delay and the rest 34% adding a delay around 0.44 seconds. On the other hand, bit.ly appears to be the slowest but shows more consistent behavior with a distance of 0.046 seconds. We speculate that this bimodal behavior of fb.me and ow.ly to be due to caching policies followed by the two services. Though, we do not observe any correlation with the time of day for either service.

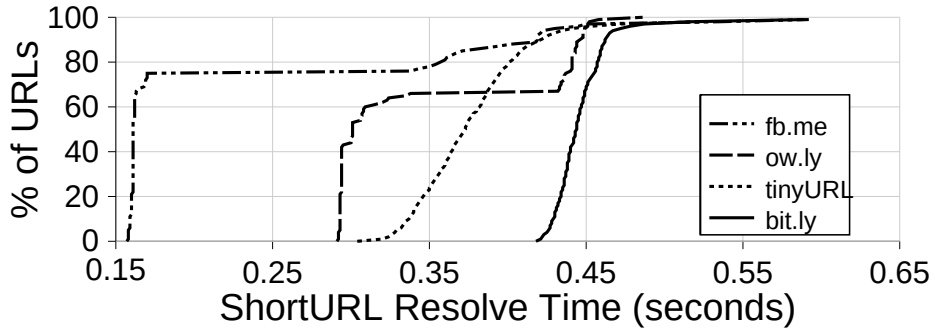


Figure 4.16: Latency in seconds imposed by 4 different URL shortening services.

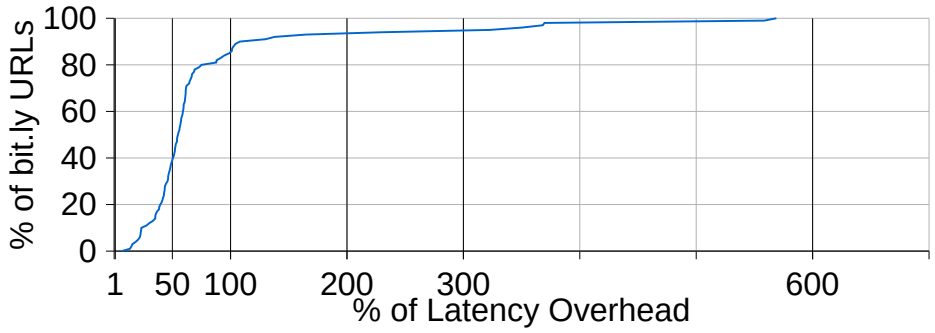


Figure 4.17: Latency imposed by URL shortening services for the 200 most popular URLs in *twitter* trace. The latency is plotted as an overhead percentage relative to the web page access time.

Figure 4.17 puts the redirection overhead of bit.ly in perspective and displays it as a percentage of the total web page access time. Using the top 200 short URLs from *twitter* we measure the additional overhead imposed for accessing a web page through a short URL. We observe that in more than 50% of the accesses, the URL shortening redirection imposes a relative overhead of 54%, while in 10% of the accesses this overhead is about 100% - a factor of two. We see then, that even though the additional delay seems to be less than half a second and may be considered small by some people, it turns out to be comparable to the final web page access time in a significant fraction of the examined cases. Therefore, should URL shortening services become even more widespread, their latency may prove even more evident, with a non-negligible penalty on performance; this implies that alternative shortening architectures for eliminating such overheads may be required in the future.

4.8 Conclusions

We have presented a large-scale study of URL shortening services by exploring traces both from the services themselves and from one of the largest pools of short URLs, namely the Twitter social network. To our knowledge, this chapter presents the first extensive characterization study of such services.

Specifically, we provided a general characterization on the web of short URLs, presenting their main distribution channels, their user community and its interests, as well as their popularity. Furthermore, we explored their lifetime and access patterns showing an activity period of more than a month with an increased popularity over the first days of their life. We explored the publishers of short URLs, and show a possibility of increased popularity when short URLs are accessed through Twitter. Additionally, a publisher of such URLs is more likely to be considered a spammer and enjoy decreased popularity when operating at an aggressive rate. Finally, we quantified the performance of URL shortening services, showing a high space gain in terms of bytes used, but also increased overhead in the web page transfer times when accessed through short URLs. This overhead increases web page access time by more than 54% in 50% of the cases, implying that alternative shortening architectures may be required in the future.

Chapter 5

Mor: Monitoring and Measurements through the Onion Router

In this chapter we present MOR, a technique for performing distributed measurement and monitoring task using the geographically diverse infrastructure of the Tor anonymizing network. MOR, was introduced in Section 3.6.4 where it was used to study the content replication in One-Click Hosting services. Based on that work, this chapter enhances the applicability and value of MOR in revealing the structure and function of large hosting infrastructures and detecting network neutrality violations. Our experiments show that about 7.5% of the tested organizations block at least one popular application port and about 5.5% of them modify HTTP headers.

5.1 Overview

A common request of researchers, administrators and simple users, is easy access to a number of geographically distributed machines. Such access would facilitate experimentation and better understanding of several network configurations, or checking for network neutrality violations. In this extent a freely available, distributed monitoring and measurement platform

would be of great value.

For many years now researchers have been using the Planetlab [CCR+03] infrastructure for conducting distributed experiments. Planetlab offers access to machines located in many different educational institutions around the world. Through a Unix-like system it allows its users to run experimental code on these machines. In this way, researchers are able to run distributed programs in many different locations, and check the network communication of their applications in real network environments. Being a (mostly) educational infrastructure, Planetlab omits commercial networks and Internet Service Providers (ISPs) with very different policies, configurations and infrastructures. Furthermore, access to Planetlab is limited to researchers. Administrators wanting to check recent configuration changes, and end users aiming at checking the quality of the service they pay for, lack a geographically distributed system freely available for this kind of daily experiments.

In this chapter we present MOR, a technique for performing geographically distributed monitoring and measurement experiments. MOR utilizes the infrastructure of the Tor anonymity network [DMS04]. Tor is a free software aiming to protect the privacy of its users, by directing users' traffic through a distributed infrastructure. This infrastructure is built by voluntarily deployed nodes in organizations, institutions and homes. To support our idea we provide a proof-of-concept implementation of such a monitoring and measurement technique. Using our technique we examine a number of case studies that show the applicability and value of MOR. The provided case studies range from examining the structure and function of large hosting infrastructures and detecting network neutrality violations. Our main contributions can be summarized as follows:

1. *We propose a technique for performing large-scale distributed measurements using the Tor anonymity network.*

2. We *demonstrate the feasibility of our technique* by providing a *proof-of-concept implementation*, and provide several different *use cases*.

3. We explore the extent of *port blocking* by various organizations over the Globe. Our results show that 11 out of 149 tested organizations (7.5%) block outgoing connections

on ports of widely used services such as ftp(21), ssh(22) and telnet(23).

4. We explore the extent to which organizations *alter HTTP headers*, and find that 9 out of 166 tested organizations (5.5%) alter, suppress or add HTTP headers.

5. We explore the extent of *Skype blocking* by organizations. Interestingly enough, we find one ISP which consistently blocks Skype.

5.2 The Tor Network

Tor [DMS04] is currently the most widely deployed anonymous communication system, with an estimation of more than 100,000 daily users around the globe [MBG+08]. These users range from ISP clients, military and company employees, to journalists and law enforcement officers [Kar09]. Used mainly for web traffic, Tor is carefully designed in order to provide anonymity for low latency services.

Tor is based on the idea of Onion Routing [GRS96], with the approach having its roots in the idea of Mix Networks proposed by Chaum [Cha81] in early 1980s. Onion Routing is built on the concept that a message from a source to a destination will first travel via a sequence of arbitrary selected proxies (*Onion Routers*). In Tor this sequence (*circuit*) is selected at random when a connection request is received (*stream*). The last node in the circuit, called an *Exit node*, is the one that will perform the actual communication with the service of interest on behalf of the user. Before the source node transfers any message to the system, it will first encrypt it with the public key of all the intermediate proxies, creating a succession of layers like an *"onion"*. Any intermediate node will then decrypt the message, with its private key, and pass it on to the next node of the circuit. When the Exit node decrypts the message, it will have the actual request data and will be able to proceed by communicating with the requested service. The response will follow the same procedure (onion creation and "un-peeling") and reverse path back to the client.

In order for Tor to succeed in providing acceptable and efficient anonymity for its users, it needs to create an overlay with a large number of proxies. In this way, latency is minimized due to the even distribution of the load to the proxies, and anonymity is improved due to

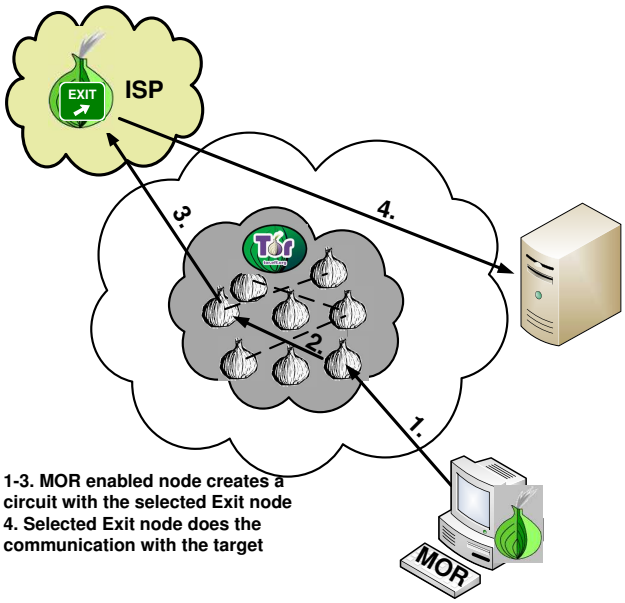


Figure 5.1: Basic steps for routing experimental traffic through the Tor network.

the wide combination of nodes for building circuits [DM06]. From the information provided by the directory servers of Tor about 1600 nodes were connected to the overlay in mid September'09. Almost half of the nodes (660) were accepting to forward traffic to the outer Internet, functioning as *Exit nodes*. These nodes were located in 48 different countries all over the word and registered by 312 different Autonomous Systems (AS). Such a large and geographically diverse overlay implicitly offers free access to the Internet through a large number of different operational networks in an easy and publicly available form. The work presented in this chapter gives a first, to our knowledge, approach of using Tor for performing monitoring and measurements tasks.

5.3 Using Tor as a monitoring and measurement platform

Participation in the Tor network is freely available and minimal effort is needed to install and configure a proxy. Though, to be able to perform experiments using Tor's infrastructure one needs to properly instrument circuit creation and connection attaching. Fortunately, the Tor community provides extended documentation for the proxy control protocol and a python library for instrumenting the proxy [The].

For any given application we want to run through Tor a basic sequence of steps has to be followed. First we select the "experimental-node-set", a proper set of Exit nodes fitting the requirements of the application. Consider a web-based application, the experimental-node-set will exclude all Exit nodes blocking outgoing traffic to port 80 in their configuration policy. After we have this experimental-node-set we follow the sequence of steps shown in Figure 5.1, iterating over each different Exit node in the set. First we create a circuit with the selected Exit node (steps 1-3). Since Tor does not allow for single-node circuits, this circuit includes at least one additional proxy. This intermediate proxy can be arbitrary selected, though we prefer to select a stable (based on its given status) router, to minimize any interference to our experiment. After the circuit is properly created, we proceed with creating the data socket for handling the required communication stream. On the first packet, the stream is attached to the previously created circuit. After these two steps are successfully completed, all traffic of the data stream is routed through the selected Exit node and our monitoring application can proceed as it would in the absence of Tor. Thus, it will send any data and wait for the response. The target host can be any online host in the Internet able to respond to our request, or a controlled host in a laboratory that would be instrumented to respond to our requests and/or log any incoming traffic from the Tor network for post analysis.

5.4 Case Studies

In this section we present a number of applications, that show the applicability and value of using Tor as an experimental platform. In our case studies we use MOR to reveal the structure and function of large hosting infrastructures and to detect network neutrality violations.

5.4.1 Examining content replication in a One-Click Hosting Service

Here we revisit our first case of interest for MOR used in Section 3.6.4, in order to understand how files are replicated and served by *www.rapidshare.com*.

Recall that, since the files shared by users of *rapidshare.com* are several MBytes large, our interest is to understand how the service replicates and serves each file to its users. To

explore this, we access the same file from many different locations (Tor Exit nodes) in order to derive the actual server(s) that provide the file each time. In our experiments we build a list with more than 20,000 *Rapidshare* URLs, available on the web, and repeatedly requesting them for download, by a group of 421 different *Exit nodes*. In this way, if server selection is done based on the client's IP address the address of the Exit node would be used by the decision algorithm.

Our experiments show that, although we observe more than 5,000 Rapidshare server IP addresses in total, *each* file is hosted only by *exactly* 12 *Rapidshare* servers. Our download attempt is always redirected to a single indexing server providing us with download URLs that point to 12 different servers hosting the actual file. Furthermore, we observed no ISP specific policy decisions, since all download requests for a specific URL where (almost) equally distributed among the 12 download servers.

5.4.2 Network Neutrality

An important discussion regarding human rights on network access is the one related to network neutrality. Internet users expect neutral treatment of their traffic from their provider, regardless the application protocol, port number or content they aim to access. In that extent, we use MOR to present a number of use cases able to infer whether a user is receiving neutral treatment from her network provider. Note, though, that use of the described experiments is not limited to end users, but an administrator can also exploit the same setup to test network or firewall configurations.

Port Blocking

Our first case study, regarding network neutrality, explores whether an organization blocks outgoing traffic from specific port numbers to the global Internet. We use a MOR client issuing access requests (TCP-SYN) to a controlled machine, located in our organization, for a number of different TCP ports. The controlled machine logs all incoming traffic using *tcpdump*, and is set out of the firewall of our organization, thus able to receive and respond to any incoming request.

Description	Port	Blocked (%)	Description	Port	Blocked (%)
Sun-RPC	111	13.16	MS-SQL	1434	7.97
Telnet	23	7.38	IMAP	143	6.70
MySQL	3306	6.52	Unreal-Game	7777	6.52
Netmeet	1503	6.47	Shiva-VPN	2233	6.47
SNMP	161	6.43	FW1-VPN	259	6.43
Netmeet	1720	6.43	Bay-VPN	500	6.38
SSH	22	6.36	Skype	5060	6.34
FTP	21	6.33	DNS-Xfer	53	6.25
IMAPS	993	5.98	HTTP	80	5.83
HTTPS	443	5.73	POP3	110	5.43

Table 5.1: Checked port numbers.

In our experiments we use the port numbers defined as *"Ports of Interest"* in [BBB07]. These ports span a large number of applications (web, p2p, e-mail, games, chat etc.). The full list of ports we use, and the description of each port, are depicted in Table 5.1. We probe each port 10 times from 236 different Tor *Exit nodes*. We consider a port to be blocked by an organization only if no TCP-SYN was received in any of the 10 tries.

Column *"Blocked"* of Table 5.1 shows the percentage of nodes that seem to be blocking each tested port number. Note that we currently have no indication whether the blocking is done by the Exit node hosting organization or somewhere in the path between that organization and our controlled machine. This kind of identification is left for future work. In all cases at least 5% of the nodes employ port blocking. From the results, we can see the largest percentage of blocking (more than 7%) to be for ports that are prone to Internet attacks, like MS-SQL and MySQL default ports (1434 and 3306 respectively) and Sun-RPC (111). Furthermore, it is interesting to see that a number of the tested organizations (more than 6%) block access to widely used services such as ftp (21), ssh (22), and telnet (23). We speculate that port blocking is done both for security considerations (ssh, telnet) and traffic discrimination (skype). Figure 5.2 plots the number of organizations (ASNs) exploiting port blocking per country.

As a further step, we examined the same port numbers using the Planetlab infrastructure [SPBP06, CCR+03]. We use 191 different Planetlab nodes, again sending 10 TCP-SYN connection requests for each port. Figure 5.3 shows the comparison between the two infras-

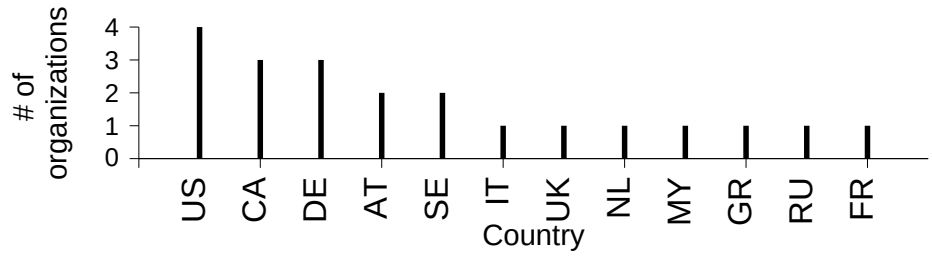


Figure 5.2: Number of port blocking organizations per country

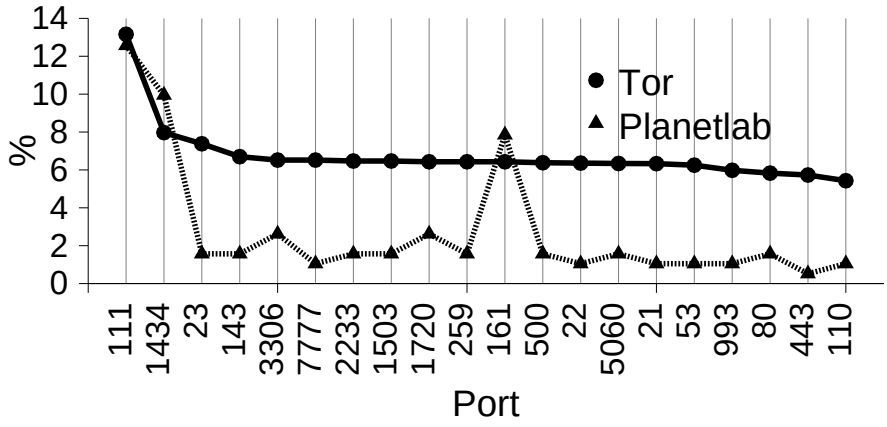


Figure 5.3: Port blocking comparison between Tor and Planetlab

tructures. We can observe a similar percentage of blocking only in three port numbers (111, 1434 and 161), which are considered prone to Internet attacks. In all other cases we observe a larger percentage of blocking in the case of the Tor overlay. Comparing the autonomous system numbers (ASN) hosting Planelab and Tor nodes we found only 10 common ASNs. Most of the organizations hosting the planetlab node were universities and research institutions. On the other hand, the percentage of academic organization in the Tor ASNs was less than 2%. Thus, MOR, by utilizing the Tor infrastructure, provides access to nodes located in commercial providers.

HTTP Header Suppression

In the next use case, we use MOR to study the suppression of *HTTP Headers* performed by different organizations. As shown in [BJM08] most of the time *HTTP Header* suppression comes from the network and not the browser, since some organizations may use intermediate

'Accept-language': 'en-us'
'Accept-encoding':
'gzip, deflate, compress;q=0.9'
'Host': '139.91.70.22'
'Accept': 'text/html,
application/xhtml+xml,
appplication/xml;q=0.9,*/*;q=0.8'
'User-agent': 'Mozilla/5.0 (X11; U;
Linux i686; en-US; rv:1.9.0.3)
Gecko/2008092510 Ubuntu/8.04
(hardy) Firefox/3.0.3'
'Accept-charset':
'iso-8859-1,utf-8;q=0.7,*;q=0.7'
'Connection': 'Close'
'Referer': '139.91.70.81'
'Cache-control': 'max-age=0'

Figure 5.4: Actual Request HTTP Header

'Content-Length': '175'
'Accept-Ranges': 'bytes'
'Server': 'Apache/2.2.3 (Debian)
DAV/2 SVN/1.4.2
mod˙python/3.2.10 Python/2.4.4
PHP/5.2.0-8+etch13
mod˙ssl/2.2.3 OpenSSL/0.9.8c
mod˙perl/2.0.2 Perl/v5.8.8'
'Last-Modified':
'Sat, 11 Apr 2009 10:05:05 GMT'
'Connection': 'close'
'etag': '"44540-9e-9d84b640"'
'Date':
'Sat, 18 Apr 2009 12:34:24 GMT'
'Content-Type': 'text/html;
charset=UTF-8'

Figure 5.5: Actual Response HTTP Header

proxies, and add or remove some headers, for caching, security and privacy reasons.

In our setup, we use a monitored machine, running an Apache HTTP server on port 80. Our MOR client issues HTTP requests for a simple web page hosted in our server. The initial *HTTP Header* contained in the request is shown in Figure 5.4. The HTTP server always responds with the header shown in Figure 5.5. [1] We used tcpdump to capture both the traffic sent and received from our client to the Tor proxy and also the traffic to and from the web server.

Our experimental-node-set contains 166 different exit nodes. In the largest percentage of Exit Nodes we observed no difference in both request and response headers. In 5.5% of the cases, though, we had suppressed headers, addition of extra header fields and in some cases responses without even accessing the web server, probably due to caching of a previous identical request from another Exit Node in the same network.

In most cases the altered, added or suppressed field reveals the existence of a proxy, either used as a centralized HTTP access point for an organization or for content caching purposes. In such cases we observe altering of the *"Connection"* header field in the request header, and addition of the *"Via"* header field in the response header. Furthermore, we observe cases where the organization completely removes the *"Referer"* or *"User-agent"* header, probably due to privacy policies. The organizations for which we observe alteration in the HTTP header fields are located in several countries, namely France, Germany, Argentina, USA, Canada and

[1] Note that in case of a difference in the HTTP Request the server will also respond differently.

China.

As a further step we run our web server on an arbitrary port number (20000). This experiment investigates whether an organization uses Deep Packet Inspection (DPI) to recognize any HTTP traffic, or whether the identification is based only on well known port numbers. In our results no altering or suppression was observed when accessing the server in a different port. Thus we can say that all used organizations do not employ DPI for HTTP traffic altering. As future work we plan to extend this study to identify traffic discrimination for file-sharing applications, like BitTorrent, through DPI.

Skype censorship

Another issue of interest, regarding network neutrality violations, arises when an organization is restricting access or limiting the performance of an Internet application, based on the port numbers used, employing DPI techniques or specific host blocking. As an example, in this case study, we explore the extent to which organizations block access to the Skype IP telephony application.

Skype utilizes a p2p infrastructure to connect its clients. When the Skype application starts it first communicates with a centralized login server (`ui.skype.com`) which will verify the user's credentials and allow her to log in to the p2p network. After the log in phase, the user's Skype traffic, either chat, voice or video, is transferred through the application's p2p overlay [BS06].

The login phase is done over the HTTP protocol by requesting a URL from the server. The URL contains the hashed user credentials and information about the running version of the application. We use this login method in order to identify organizations that block Skype users from logging into the system. For our experiments we extracted the URL from the latest version of a Linux Skype client. [2] We request the URL from Skype's login server through Tor, using 171 different Exit nodes. For every Exit node we request the URL 10 times and log the response result. We consider the organization not to be blocking access to Skype if we receive at least one valid response from the server. Our connection requests were

[2] `http://ui.skype.com/ui/2/2.0.0.72/en/getlatestversion?ver=2.0.0.72&uhash=1074a31ab9146cc11ab149c86a32dc920`

100% unsuccessful in only one case. This Exit node is hosted by a Kuwait ISP which has also been reported by others to block Skype. [3]

5.4.3 Further Possible Use Cases

Web-Page Censorship: Recent work has shown that a number of web clients receive altered pages during their browsing sessions [RGKW08]. These alterations may include advertisements, extra javascript code and even malware, that are either annoying (in the best case) or harmful to the user. Using our technique one can compare the page she receives from a web server, with the pages received when the server is visited from a different geographical location and/or ISP.

Network Problems Diagnosis: Using MOR one can easily detect if an administered or desired service is working properly. For example a user can check if a service is non responsive also from other organizations or only from it's own network in order to report this to her administrators. Also online service administrators can check the visibility and correct functioning of their service when viewed from external networks.

DNS update speed: Using a SOCKS4a proxy one can direct DNS queries through the Tor network. Combined with our technique one can measure the time needed for a domain name update to become visible by the rest of the Internet.

5.5 Discussion, Limitations and Conclusions

In this chapter we propose a new technique for performing measurement and monitoring tasks. We propose the use of the Tor anonymizing network as a geographically distributed infrastructure. Using our proof-of-concept implementation, MOR, we present a number of case studies that demonstrate the applicability and value of our approach. Our experiments show that about 7.5% of the tested organizations block at least one popular application port and about 5.5% of them modify HTTP headers.

While our work actually leverages the Tor network, the applications we propose make careful use of the network adding limited overhead (i.e. single TCP-SYN packets and small

[3] http://www.248am.com/mark/kuwait/skype-blocked-by-qualitynet/

web requests). We expect MOR to act as a motivation for a number of users, interested in measurement and monitoring tasks, in adding more relays to the network. In this case, Tor will benefit from MOR users, since more proxies will increase the network's geographic diversity, improve anonymity and Tor's overall performance [ADS03].

Limitations: Unfortunately, with the current state of the Tor network, a number of interesting tasks can not be implemented. Two main limitations are the lack of relaying non-TCP traffic and the limited throughput performance. Tor does not, for the time being, support relaying non-TCP traffic. In this extent a number of programs (i.e. traceroute) that use other IP protocols, can not be relayed through Tor. Due to this, a number of experiments based on this type of tools are not feasible. Furthermore, though Tor tries, and succeeds, to provide low latency anonymization, it is still not able to support high throughput applications. In this case experiments targeting on identifying path delay, loss and average/peak throughput are not guaranteed to provide accurate results. It is highly possible that the measured metric will be affected by the system itself and will not correspond to the actual value from the targeted network. Though these are fundamental limitations on the number of possible use cases of our technique, we believe that our work will encourage exploration for integrating the aforementioned metrics.

Chapter 6

Identifying Web User Activity Sessions

Web 2.0 applications often require longer interaction times with the user, either through explicit actions on behalf of the user or through automatic updates of the Web page. Traditional methods for identifying Web Sessions collect bunch of HTTP(s) request-response pairs separated by the estimated "think" time of the user. In the Web 2.0 era these methods fail to collect this continuous activity of the Web pages in a single session for each application. In this chapter we present a new methodology for identifying Web Activity Sessions. These sessions include the total activity of the user with a Web application/page and give an insight to the behavior of Web users in the Web 2.0 era. Our identification method relies on the following important observation regarding Web traffic. A Web Activity Session is a set of flows between a client host and a certain set of server IP addresses. These addresses may belong to different prefixes, but it is quite unlikely that the "core" addresses, or prefix, serving a specific service will change during an application session with the user.

We evaluate our approach using a number of network traces obtained during a user study that took place in our laboratory. Our approach is able to identify the activity sessions reported by the users during the study. Furthermore, it also identifies automatic activities coming from the browser or the Operating System, like updates and security checks.

6.1 Introduction

As mentioned in Chapter 1 the evolution of Web 2.0 during the past years has brought a number of changes in the usage, development and deployment of Web pages. As a result, the Web has evolved from simple Web pages and static images to a number of Rich Internet Applications. Many applications that traditionally lived into the operating system, as a separate program, are now available through the Web Browser window. From email to rich document editors and from video viewing to live video streaming, all these applications have made the Web Browser to be through as the new Operating System [WGM+09]. The aforementioned applications, take advantage of new Web technologies, like AJAX [imp05], to provide a more dynamic and realistic environment (compared to their desktop alternatives). This results to Web applications becoming more active; continuously polling for updates and new information of interest for their users. Furthermore, with all major Web Browsers allowing multi-tab capabilities, multiple applications are operating at the same time, exchanging data with a number of Web Servers.

This changing and evolving Web clearly affects user's behavior and interaction with the Web pages. In result, Web Sessions do not further consist of just downloading and viewing a Web page, but also include a number of asynchronous data exchanges. Traditional approaches for identifying Web Sessions are based on the ON/OFF activity of a Web user. These approaches cluster user's HTTP(s) request-response pairs based on their time difference (ON time). These clusters are then splitted into Web Sessions based on the estimated "think" time of the user (OFF phase). By design, these methods will fail to cluster together activities of the same application. Request-response pairs belonging to the same application, but with a time difference larger than the estimated "think" time, will result to different sessions.

In this chapter we define Web Activity Sessions. [1] We refer to this term as the network data exchanged from a Web user during the lifetime of a single activity. Activities range from a single video view or file download, to the use of a web-mail application, performing web-based chat or browsing a news-page. Our goal, for Web Activity Sessions, is to capture the

[1] We use the terms Web Sessions, Sessions and Web Activity Sessions to refer to Web Activity Sessions throughout the text.

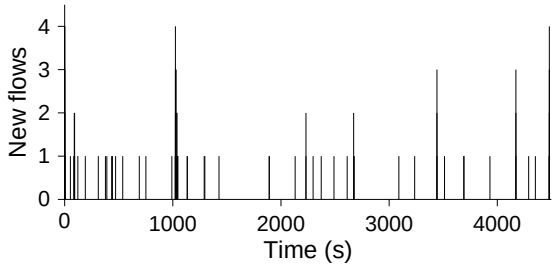
total interactions of a user with a Web application/page during a single activity. Identifying these Web Activity Sessions is just the first step for analyzing Web traffic in this new Web era. From this analysis we aim to (1) understand how new technologies and Web application have changed Web user behavior and usage, (2) create a new model for Web traffic able to capture these changes and assist in the evolution of Web technologies and infrastructures, (3) provide a Web Activity identification method able to capture the core activity of the users in each Session.

The remainder of this chapter is structured as follows. Section 6.2 gives the main ideas and definition for the Web Activity Sessions. Section 6.3 describes our identification method and provides an evaluation using our user-annotated network traces. Finally, Section 6.4 concludes this chapter and presents future plans.

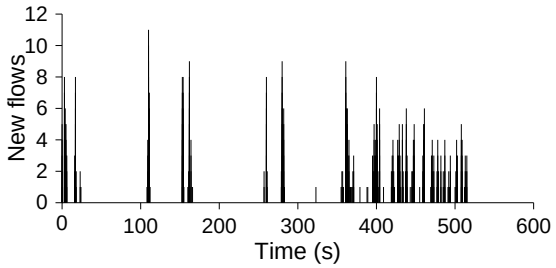
6.2 Web User Activity Sessions

Traditional approaches identify web sessions based on the ON/OFF activity of a Web user. In this model the ON activity is consisted by a number of consecutive HTTP/HTTPS request/response pairs from the same client to a number of servers. To separate different web sessions from the same client, these methods use an interarrival heuristic that corresponds to requests of the same session (if the gap is less than 1 second) or the user's "thinking" time (OFF time), if the gap is in the range of a few minutes [Mah97].

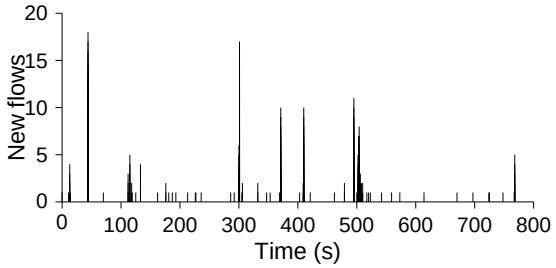
In the traditional Web model this type of session identification is limited only by the definition of the interarrival heuristic. It is able to identify most of the browsing sessions. In the emerging web era though, we are experiencing a number of "interactive" applications. This interaction may come from a User's action or a browsers automatic request. Most of the time this requests are related to the current action of a user and should not be considered as a different activity. As an example, consider a user viewing streaming video through it's browser. The browser plugin (and hence the browser) will continuously request a new video chunk every few seconds. If this request period is larger than the interarrival heuristic it will result in identifying a number of web sessions instead of a single one.



(a) Gmail



(b) CNN



(c) Facebook

Figure 6.1: Examples flow starts for different web applications.

In this chapter we try to overcome this limitation of previous Web session identification techniques. We define the notion of Web Activity Session to describe the set of flows that contribute to a single activity of a user. As an activity we define the interaction of a user with a specific service. Examples of these services might be the browsing of a static web page, using Web mail, having a chat conversation or viewing a video.

Our identification method is based on two basic observations present on web traffic. First, due to the varying parts consisting a web page (content, ads, statistic scripts), every activity of a user will result in a number of new flows responsible for the download of the content. Second, a single activity will be served by a limited number of servers. Furthermore, is very

unlikely that the core servers responsible for this activity will also be responsible for a different activity.

The first observation helps us identify the start time of an activity, while the second one helps to attributing sporadically appearing flows to the activity they belong to. Figure 6.1 presents the observed new flows per second for some real world examples of these Web Activity Sessions. Figure 6.1(a) presents interactions of a user with Gmail. New flows in consecutive seconds are observed with darker lines. We can see a number of single flows appearing sporadically. These flows are automatically produced from the application in order to check for new mails and are highly correlated to the activity of the user during the session. Other activities like login (sec 60), send and receive mail (around second 1000) and the beginning of a chat conversation (around second 2600) can be also observed due to the number of new flows. Figures 6.1(b) and 6.1(c) present similar examples for visiting cnn.com and facebook.com. We can observe the new flow peaks appearing during viewing a number of different articles in cnn.com (around seconds 0, 20, 120, 180, 300, 360) and also during the viewing of a photo gallery starting at second 420 and continuing with chunks of new flows for every requested new photo. Similarly for facebook.com we observe a photo view activity around second 120, browsing several profile pages (around seconds 300, 360 and 400) and the beginning of a flash game at around second 500.

6.3 Identifying Web Activity Sessions

In this section we describe our approach for identifying Web activity session. We first give a detail presentation of the designed method. Following, we present results from our evaluation with user annotated packet traces.

6.3.1 Methodology

Our implementation uses TCP/IP headers from offline network packet traces. Since we are only interested on Web traffic we use only packets originating and destination from and to TCP port 80 (HTTP) and TCP port 443 (HTTPS). Although non-Web applications may

also use these port numbers to exchange traffic and evade firewall and security restrictions, Kim *et al.*in [KCF+08] provide evidence that only 0.1% - 0.5% of the flows on these ports belongs to applications different than Web. We consider our port selection safe enough for performing this analysis.

We separate network flow information as identified by the 5-tuble field, Source and Destination IP addresses, Source and Destination ports and protocol. We keep detail information for all communication flows originating from every IP address. This flow information includes the start and end times of the flow, the number of packets exchanged for both directions, as well as the number of data packets and bytes exchanged. Data packets define both the activity period of a flow, and the consumer/producer relationship between the members of the communication.

After we collect the information for all flows from a client during our observation period we proceed with identifying the activity sessions for that client. First, we employ a cleanup procedure to our flow-set. During this procedure we remove all the flows that either have no activity (i.e. no data packets send or received from the client) or flows for which we have not seen the TCP connection establishment. The former contribute zero activity to a session and most of the time are the result of misconfigurations or malicious activities that we consider as noise. Flows that were active before the start time of our observation period are removed from the flow-set in order to have a clean state session identification.

The second step aims at identifying groups of flows that are initialized consecutive. The intuition behind this step is that a single user interaction with it's Web browser is translated to more than one network interactions. Usually parts of the web page are hosted into a number of different Web servers. Thus, a single user click will result to multiple flows being initialized consecutive. Passing through the flow records for a client we group together all flows that where established during the same or neighboring seconds. The resulting groups contain all the flows that were established during a single user action (i.e. a click on a URL or sending an email).

The final step for identifying the Web Activity sessions is based on the observation that a single activity will be served by a limited number of core servers. These core servers, if not

a single one, will very likely belong to the same subnet. Thus we compare each flow group with all other flows in a window of 100 seconds (before and after the start of that group). If we find a group that has a large overlap in the number of flows or bytes from the same server or /24 subnet we then combine these two groups. In case we find multiple candidate groups, we combine the group with the larger overlap. We iterate this procedure until no overlaps are further found.

6.3.2 Evaluation

To evaluate our Web Activity Session identification approach we performed a user study, requesting from a number of users to log down their Web activities start and end times. For the study we used a single machine connected to the Internet. Each user was requested to use that machine for a variable time period in order to perform her everyday Web tasks. The study was performed for 4 days from 12 users. During this time the users produced almost 2 GBytes of Web traffic during more than 12 hours of browsing time. Each user was asked to report the time she consider that she changed her activity during her Web interactions. For further reference the user were requested to report the type of activity they were performing during that time period. The most reported activities where Browsing (representing traditional web browsing), Mail, Video and Image viewing. Other activities reported where file Downloads, Chat sessions and use of Interactive applications.

As a comparison measure we also implemented a traditional Web session identification method, similar to the one described in [PD07]. This method works as follows: we first identify the different transfers observed in each TCP connection, defined by the unique 5-field tuble (Source and Destination IP address, Source and Destination Port and Protocol). A transfer is identified by the packets within a single connection that have inter-arrival values smaller than 1 second. This transfer partitioning allows as to identify different data transfers that occur in a persistent TCP connection, after some "view" time on behalf of the user.

Usually a user's action will result to several transfers. Since these transfers are automatically generated by the browser, their inter-arrival time will be much lower than the latency of a user action. We use a threshold of 40 seconds in order to group a number of transfers in

User Reported Activities	Traditional Web Sessions	Web Activity Sessions
224	2771	309

Table 6.1: Evaluation of Web Activity Session Identification.

the same web session.

Table 6.1 provides the results of our evaluation. During the study the users reported 224 different activity sessions. The traditional Web session identification method reported an order of magnitude more session than the ones reported by the users. On the other hand our method identified about 80 more session. There are several reasons for the difference in the number of sessions identified. First, we should note that all latest browsers automatically load a home page at start time. This page was not reported by the users as an activity because it was not consider one. Furthermore, most of the browsers include blacklists for performing security checks on hostnames and URLs for phishing and malware attacks (i.e. google's safebrowsing). This blacklists are continuously and automatically update by the browser, thus creating a number of extra sessions not reported by the user. A third case for extra sessions created are some statistic plugins added by web page developers for collection information about their web page visits and browsing times. These services are usually hosted by third-party entities and upload information asynchronously. Most of these information are uploaded after the full load of the web page and often result in a different session by themselves.

6.4 Conclusions and Future Work

With the evolution of the Web Browser as a real-time application platform Web Sessions involve much more interactions than single page and image downloads. In this chapter we presented a method for identifying Web Activity Sessions. These type of sessions capture the total interactions of a user with a Web application during a single activity. Using a number of annotated user traces, collected during a user study performed in our lab, we showed the ability of our method to capture the activities of the users.

We consider our work as the first step for analyzing Web traffic in this new and evolving

Web era. Our ongoing work targets a more thorough analysis of Web Activity Sessions. From this analysis we aim to (1) understand how new technologies and Web application have changed Web user behavior and usage, (2) create a new model for Web traffic able to capture these changes and assist in the evolution of Web technologies and infrastructures, (3) provide a Web Activity identification method able to capture the core activity of the users in each Session.

Chapter 7

Related Work

In this chapter we place our work in the appropriate context by presenting work related to the content of this dissertation's research efforts.

7.1 Peer-to-Peer File Sharing

File sharing has attracted significant interest over the last ten years through the emergence of p2p protocols. The effect of p2p networks has been studied extensively [Mar02, NRS+02, Coh03, IUKB+04, GCX+05]. Saroiu *et al.* studied the characteristics of participating peers in Napster and Gnutella, and observed significant heterogeneity and lack of cooperation among them [SGG03]. Sen and Wang analyzed the characteristics and bandwidth requirements of p2p traffic in large networks [SW02]. Leibowitz *et al.* examined file popularity in the Kazaa network, and its effect on the total traffic volume [LRW03].

The work of Gummadi *et al.* examined Kazaa's and Gnutella's file sharing workloads, and revealed significant popularity deviations between objects shared in kazaa and web objects mostly due to the "fetch-at-most-once" behavior of Kazaa users [GDS+03]. Karagiannis *et al.* on the other hand, showed that ISPs can benefit, in terms of cost, from locality-aware p2p distribution [KRP05]. Our results for OCH services, Chapter 3, suggest that caching will not benefit ISPs since 75% of the files in our 4-5 month traces were downloaded only once.

7.2 Web 2.0 File Sharing

Several recent studies focused on emerging web 2.0 applications that host user-generated content. Gill *et al.* studied the characteristics of YouTube users' content in an academic network [GALM07]. Cha *et al.* also studied user-generated content sites by crawling YouTube and Daum [CKR+07]. Focusing on video popularity, both studies found that the YouTube video workload is similar to traditional web workload, thus end-users and organizations can benefit in throughput and reduced bandwidth demands, respectively, by employing caching policies or distribution through p2p systems. In late 2008 Cho *et al.* observed a slow but steady movement of residential users towards rich-media content, like YouTube, by comparing data from 2005 and 2008 [CFEA08]. During the time period of our study on One-Click Hosting services we did not observe a significant increase in OCH traffic, but we expect users to gradually turn to such services because they often offer better content availability and higher throughput than p2p networks.

7.3 Content Distribution Networks

Work on CDNs, like Akamai [aka], has focused on the evaluation and improvement of the redirection and server selection algorithms, which result in lower latency and better quality-of-experience for end-users. Krishnamurthy *et al.* showed, in 2000, that the use of multiple distributed servers and caching in content distribution can improve end-to-end web performance [KW00]. Johnson *et al.* measured the performance of two content distribution networks, Akamai and Digital Island, built over DNS redirection [JCDK01]. They note that both systems provide good but not optimal performance and conclude that their goal is not to select the best server, but to *not* select a bad server. Krishnamurthy *et al.* proposed a methodology to study the client-experienced performance using CDNs, and also performed an extensive study of the use of such systems [KWZ01]. The work presented by Saroiu *et al.* pursued a trace-based characterization of four Content Delivery Systems, Gnutella, Kazaa, WWW and Akamai [sar02]. Their results showed that Akamai consumed only a minimal amount of traffic (0.2%), while Gnutella, WWW and Kazaa followed with 6.04%, 14.3% and

36.9%, respectively.

7.4 Online Social Networks

Interest in online social networks and services has been significant over the past years. Several measurement studies have examined basic graph properties such as degree distributions or clustering coefficients [CKE+08, MMG+07] or their particular structure [KNT06]. While part of our traces originates from Twitter, our work significantly differs from these studies as we focus on the use of short URLs and their presence within a social network, rather than network itself (see Chapter 4).

Part of our analysis in Chapter 4 relates to the evolution of content popularity [CMAG08, CMG09], information propagation through social links [CMG09, LAH06], as well as popularity of objects and applications in social networks [CKR+07, NRC08]. For example, in [CMAG08, CMG09] the authors study how Flickr images evolve and how information propagates through the Flickr social graph. Lerman and Ghosh in [LG10] examined the information spread in Twitter and Digg and showed that although Twitter is a less dence network and spreads information slower than Digg, information continues to spread for longer and penetrates further the social graph. In a spirit similar to these studies, we examine how content becomes popular over time. However, in our work, we focus on how this popularity is reflected by the hit rate of short URLs. Cha et al [CKR+07] also deal with content popularity by performing a study of user generated content via crawling the YouTube and Daum sites. The authors observed the presence of the Pareto principle. Our analysis confirms that this is also the case in the popularity of short URLs. Our observations on the dispersion of the hit rates of short URLs are consistent with the well-documented findings on the existence of Zipf's Law and heavy-tailed distributions in WWW (e.g., [BCF+98, CB97]). However, our work further highlights that a web site's popularity does not necessarily translate in an equivalent popularity in the "web of short URLs".

7.4.1 Information Propagation in Twitter

Information propagation in Twitter has been studied in [KLPM10]. The authors have crawled the Twitter network and analyzed the temporal behavioral of trending topics. The authors suggested that Twitter is mostly a news propagation network, with more than 85% of trending topics reflecting headline news. Indeed, this observation is also confirmed by study on short URLs. A large fraction of short URLs points to news-related domains; however, the percentage of news related URLs appears lower in our study, 7 out of the top-100 URLs.

7.5 Mis-using Overlay Networks

With Tor being increasingly popular during the last years, a number of researchers examined the network, targeting its performance [SB08], attacking the system [BDMT07, PAIM08]. or trying to compromise its users' anonymity through traffic analysis [MD05, EDG09].

The goal of performing a variety of distributed experiments, led a number of researchers to use overlay networks from a different aspect than the one initially intended. Athanasopoulos *et al.* in [AAM06] used the Gnutella overlay network to perform Distributed Denial of Service attacks on third party services. In a subsequent work the same authors illustrated the use of Gnutella in anonymously downloading a web file [ARAM07]. Close to our work, Beverly *et al.* in [BBB07] used the Gnutella Network to quantify the prevalence of port blocking from ISPs and institutions. In their setup, a super-peer in the Gnutella network was instrumented to redirect each contacting client to a specific port of a measurement host controlled by the authors. Recently, Barth *et al.* in [BJM08] used advertisement networks to study the use and suppress of the *HTTP Referer* field. They used two advertisement networks to display custom advertisements to the users for 3 days. When the advertisement was displayed in the user's web browser, it issued a number of HTTP requests to two servers controlled by the authors. Their observation showed that most of the times, the *Referer* HTTP field was suppressed in the network and not in the browser.

7.6 Web Sessions

Several studies used the notion of web session to understand Web user behavior and model HTTP traffic [Mah97, CL99, BMM+09, HCJS03, SCJO01]. Mah in [Mah97] uses a time threshold between two connections in order to identify the number of files per Web page. Prasad and Dovrolis in [PD07] used a similar approach to identify Web Sessions in order to measure congestion responsiveness of Internet traffic. We used their Web Session extraction method to identify traditional Web Sessions during our evaluation in Section 6.3.2. Choi and Limb in [CL99] used the file extension information from the HTTP header to identify Web-requests. They defined Web-requests as a page or a set of pages that results from an action of a user. Bianco *et al.* in [BMM+09] used clustering techniques to identify Web Sessions, aiming at avoiding an *a priori* definition of threshold values. All these techniques target at identifying a single user interaction with a Web page. In contrast, our Web Activity Sessions aim at grouping together user interactions and viewing the Web page as a single application. characterization

The evolution of Web traffic has also been of interest to several researchers. Li *et al.* in [LMC08] presented a preliminary classification of Web Requests using payload signatures. Their analysis from two network traces from 2003 and 2006 showed a large increase in non-Web activities over HTTP traffic. In [SAAF08], Schneider *et al.* presented a characterization of four Web 2.0 applications, showing that these applications transfer a larger number of bytes, through many more request than traditional Web pages, and also are actively pre-fetching data. Lee *et al.* presented a measurement system for the collection of AJAX related traffic aiming at the identification of individual operations performed by the user during interaction with a Web application [LKS10].

Chapter 8

Concluding Remarks

In the context of this thesis we presented a study of two widely deployed mechanisms for file and information sharing. These services emerged during the evolution of web 2.0. In this chapter we summarize the results of our study, present the impact of our work through the discussion of some follow-up work and discuss some directions for further research regarding file and information sharing in web 2.0.

8.1 Summary of Results

The first part of this thesis, Chapter 3, presented a study of One-Click Hosting services, web based file sharing services that, as shown, produce a significant share of the daily web traffic. Our study on OCH client usage habits showed that eventhough most users download more than one file per day, we observe very little locality of reference in the objects downloaded. These low locality patterns imply that there will be no significant benefits by caching content near to users. To cope with this, our experiments revealed that RapidShare employs a centralized, heavily multi-homed server infrastructure that is located at a single geographical location. Furthermore, comparing OCH services to the popular BitTorrent p2p network showed that OCH services have the qualification to become a leading file-sharing platform. Content availability is in most cases higher than BitTorrent, while free users experience at least equal download performance.

An essential component for the success of OCH services are the many blogs and wikis that act as content indexing sites. Using these sites users create communities for sharing and discovering content. We examined a number of indexing sites and found that they mostly point to copyrighted content, as is also the case with p2p file sharing systems.

In the following part of the thesis, Chapter 4, we presented a study on URL shortening services. Our study analyzed data retrieved both by crawling the services themselves and by retrieving data from the Twitter social network. Our study showed that short URLs appear mostly in ephemeral media, like Online Social Networks, suggesting a "word of mouth" URL distribution. Their access distribution suggests that only a small portion of short URLs becomes very popular, but also that more than 50% are accessed for a period larger than three months. Though not ephemeral, short URLs accesses appear in bursts, becoming popular extremely fast, suggesting a "twitter effect". Examining the top, in terms of popularity, web sites accessed through short URLs showed a profoundly different set compared to the sites popular in the broader web community. These results suggest that the communities evolved through the social web create different access patters than the traditional web. Access patterns that may pose interesting design challenges for the web sites.

Our study on short URLs also examined the performance implications of their usage. We found that in more than 90% of the cases, the resulting short URL reduce the amount of bytes needed for the URL by 95%. This result suggests that URL shortening services are extremely effective in space gaining. On the other hand, we observed that the imposed redirection of URL shortening services increases the web page access times by an additional 54% relative overhead. This result should be taken into consideration for the design of future URL shortening services.

In Chapter 5, we presented a technique for leveraging the Tor anonymizing network as a geographically distributed infrastructure for performing measurement and monitoring tasks. Through a number of case studies we demonstrated the applicability and value of our approach. Our experiments show that about 7.5% of the tested organizations block at least one popular application port and about 5.5% of them modify HTTP headers. Our technique was used also to examine the infrastructure of OCH services, in Chapter 3.

Finally, in Chapter 6 we presented a method for identifying Web Activity Sessions. These type of sessions capture the total interactions of a user with a Web application during a single activity. Using a number of annotated user traces, collected during a user study performed in our lab, we showed the ability of our method to capture the activities of the users. Web activity sessions help as to better understand web 2.0 usage and provide an important tool for network traffic identification of Web applications.

8.2 Directions for Further Research

Enabling caching in OCH services: Our observations show that OCH services have low locality patterns. This significantly affects the infrastructure organization, since it does not allow for caching content near the user and other optimizations made by the ISP in order to improve its network infrastructure, quality of service and operational costs. Furthermore, as we have seen OCH users organize themselves in multiple social communities that share common interests. Additionally, OCH services reward users in several ways when their uploaded files are download many times, giving them the incentive to continuously upload new files. All this factors may affect the number of copies of the same file/object that are uploaded to the service. A wide study of the OCH indexing services would reveal this number and would be in place to develop new methods of storage that would enable object caching near the ISP (like storing the file into smaller pieces, similarly to the way Dropbox does [MSL+11] but without the security implications).

URL shortening services might be a single point of failure: Our analysis of the Twitter dataset showed that short URLs accumulate more than 80% of the total HTTP URLs appearing in the OSN. Furthermore, the most popular URL shortening service, namely bit.ly, counts for more that 50% of the total short URLs. It is obvious that any, accidental or malicious, downtime of one such service will result in users not being able to access their information. This threat raises the need for alternative URL shortening service architectures, focused in providing access to the content through a distributed infrastructure or even by utilizing the social graph of the OSN.

What makes short URL content non-ephemeral: Our study revealed that more than 50% of the short URLs survive for periods of more than 3 months, making them not ephemeral. An interesting question, both for publishers and content providers, would be to understand what affects the lifetime of a short URL. For example, one may examine whether non-ephemeral short URLs are bind to a specific publisher of type of content. A close approach, not bind to short URLs, was examined by [WHMW11].

Appendix A

Publications

This thesis has produced several publications in peer-reviewed conferences.

The research associated with *One-Click Hosting Services* is published in the following publications:

> Demetris Antoniades, Evangelos P.Markatos and Constantine Dovrolis. **One-Click Hosting Services: A File-Sharing Hideout.** *In Proceedings of the 9th ACM Conference on Internet Measurements (IMC 2009).* November 2009, Chicago, Illinois, US.

The research associated with *URL Shortening Services* is published in:

> Demetris Antoniades, Iasonas Polakis, Giorgos Kontaxis, Elias Athanasopoulos, Sotiris Ioannidis, Evangelos P. Markatos and Thomas Karagiannis. **we.b: The Web of Short URLs.** *In Proceedings of the 20th World Wide Web Conference (WWW'11).* March 2011, Hyderabad, India.

The research associated with the *MOR* technique is published in:

> Demetris Antoniades, Evangelos P. Markatos and Constantine Dovrolis. **MOR: Monitoring and Measurements through the Onion Router.** *In Proceedings of the Passive and Active Measurement Conference (PAM'10).* April 2010, Zurich, Switzerland.

Appendix B

Impact

In this section we discuss the impact of our work as this can be represented by the variety and quality of the publications that cited our work. A list of the publications produced by the work presented in this thesis can be found at Appendix A.

Poese *et al.* in [PFA+10b] presented PaDIS, a system that assist users/ISPs/CDNs to select the best content delivery server based on an ISP specific view. Examining an OCH infrastructure similar to the one we discovered in Section 3.6 the authors identify potential for better download and upload rates when peering ISPs are used.

Liu *et al.* presented FS2You [LSL+10], a real Online Hosting Services that uses Peer-Assisted storage and content distribution in order to minimize the cost of the service itself, with the effect of sacrificing availability for less popular files.

Nikiforakis *et al.* in [NBVA+11] exposed the privacy implications of OCH services by examining the download URL generation process in more than 100 such services and showed that a predictable generation was used by close to 40% of the services. Their analysis also showed that attackers are currently leveraging this weakness to gain access to user's private information.

Rondrigues *et al.* in [RBC+11] examined the *world-of-mouth* based discovery of the web by examining the URL exchange on Twitter. Similarly to our work on short URLs they identified that the most popular domains shared on Twitter differ significantly from the general web case (See Section 4.4.4). Furthermore, they revealed that Twitter yield propagation trees

that are wider than they are deep (attributed to the design of twitter where receivers choose senders and not vice-versa) and that is likely for geographically closer users to share the same URL.

Chhabra *et al.* in [CABK11] examined the effect of phishing attacks using short URLs in Twitter. They showed that phishers target OSNs more than traditional brands like eBay and banks.

Bibliography

[AAM06] E. Athanasopoulos, K.G. Anagnostakis, and E.P. Markatos. Misusing Unstructured P2P Systems to Perform DoS Attacks: The Network That Never Forgets. In *Proceedings of the 4th International Conference on Applied Cryptography and Network Security*, June 2006.

[AANPF] Akamai Accelerated Network Program FAQ. `http://www.uvm.edu/~sjc/vbns/akamai/akamai-aanp-faq.pdf`.

[ADS03] Alessandro Acquisti, Roger Dingledine, and Paul Syverson. On the Economics of Anonymity. In Rebecca N. Wright, editor, *Proceedings of Financial Cryptography (FC '03)*. Springer-Verlag, LNCS 2742, January 2003.

[aka] Akamai. `http://www.akamai.com`.

[ale] Alexa Traffic Stats. `http://www.alexa.com/siteinfo/bit.ly#trafficstats`.

[ARAM07] E. Athanasopoulos, M. Roussopoulos, K.G. Anagnostakis, and E.P. Markatos. GAS: Overloading a File Sharing Network as an Anonymizing System. In *Proceedings of the 2nd International Workshop on Security*, 2007.

[BBB07] Robert Beverly, Steven Bauer, and Arthur Berger. The Internet's Not a Big Truck: Toward Quantifying Network Neutrality. In *Proceedings of the 8th Passive and Active Measurement Workshop*, pages 135–144, April 2007.

[BCF+98] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *IN INFOCOM*, pages 126–134, 1998.

[BDMT07] Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of Service or Denial of Security? How Attacks on Reliability can Compromise Anonymity. In *Proceedings of the 14th ACM Conference on Computer and Communication Security*, October 2007.

[Bel02] Steve M. Bellovin. A Technique For Counting NATted Hosts. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, pages 267–272. ACM New York, NY, USA, 2002.

[BJM08] Adam Barth, Collin Jackson, and John C. Mitchell. Robust Defenses for Cross-Site Request Forgery. In *Proceedings of the 15th ACM conference on Computer and Communications security*, pages 75–88. ACM New York, NY, USA, 2008.

[BMM+09] Andrea Bianco, Gianluca Mardente, Marco Mellia, Maurizio Munafò, and Luca Muscariello. Web user-session inference by means of clustering techniques. *IEEE/ACM Trans. Netw.*, 17(2):405–416, 2009.

[BS06] Salman A. Baset and Henning G. Schulzrinne. An Analysis of the Skype Peer-To-Peer Internet Telephony Protocol. In *IEEE International Conference on Computer Communications*, pages 1–11, 2006.

[CABK11] Sidharth Chhabra, Anupama Aggarwal, Fabrício Benevenuto, and Ponnurangam Kumaraguru. Phi.sh/$ocial: the phishing landscape through short urls. In Vidyasagar Potdar, editor, *CEAS*, pages 92–101. ACM, 2011.

[CB97] Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[CCR+03] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. PlanetLab: An Overlay Testbed For Broad-Coverage Services. *ACM SIGCOMM Computer Communications Review*, 33(3):3–12, 2003.

[CFEA08] Kenjiro Cho, Kensuke Fukuda, Hiroshi Esaki, and Kato Akira. Observing Slow Crustal Movement in Residential User Traffic. In *Proceeding of the 4th Annual Conference on Emerging Network Experiments and Technologies, CoNEXT'08*. ACM, 2008.

[Cha81] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[CKE+08] H. Chun, H. Kwak, Y. Eom, Y. Ahn, S. Moon, and H. Jeong. Comparison of online social relations in volume vs interaction: a case study of cyworld. In *IMC '08: Proc. of the ACM SIGCOMM conference on Internet measurement*, pages 57–70, 2008.

[CKR+07] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM New York, NY, USA, 2007.

[CL99] Hyoung-Kee Choi and John O. Limb. A behavioral model of web traffic. In *ICNP '99: Proceedings of the Seventh Annual International Conference on Network Protocols*, page 327, Washington, DC, USA, 1999. IEEE Computer Society.

[CMAG08] M. Cha, A. Mislove, B. Adams, and K. Gummadi. Characterizing Social Cascades in Flickr. In *ACM SIGCOMM Workshop on OSNs*, 2008.

[CMG09] M. Cha, A. Mislove, and K. P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *Proc. of the 18 Intl. World Wide Web Conference (WWW)*, 2009.

[Coh03] Bram Cohen. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, volume 6. Berkeley, CA, USA, 2003.

[DD08] Amogh Dhamdhere and Constantine Dovrolis. Ten Years In The Evolution Of The Internet Ecosystem. In *IMC '08: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, pages 183–196. ACM New York, NY, USA, 2008.

[DET80] S. Daniel, J. Ellis, and T. Truscott. Usenet, a general access unix network. *Duke University*, 1980.

[DiN99] D. DiNucci. Fragmented future. *Print*, 53(4):32, 1999.

[DM06] Roger Dingledine and Nick Mathewson. Anonymity Loves Company: Usability and the Network Effect. In Ross Anderson, editor, *Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006)*, Cambridge, UK, June 2006.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: the Second-Generation Onion Router. In *Proceedings of the 13th conference on USENIX Security Symposium*, pages 21–21. USENIX Association Berkeley, CA, USA, 2004.

[EDG09] Nathan Evans, Roger Dingledine, and Christian Grothoff. A Practical Congestion Attack on Tor Using Long Paths. In *Proceedings of the 18th USENIX Security Symposium*, August 2009.

[GALM07] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 15–28. ACM New York, NY, USA, 2007.

[GCX+05] Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang. Measurements, analysis, and modeling of bittorrent-like systems. In *In*

Proceedings of the ACM/SIGCOMM Internet Measurement Conference (IMC-05), 2005.

[GDS+03] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *19th ACM Symposium on Operating Systems Principles (SOSP2003)*, Bolton Landing, NY, October 2003.

[GQX+04] David K. Goldenberg, Lili Qiuy, Haiyong Xie, Yang Richard Yang, and Yin Zhang. Optimizing cost and performance for multihoming. *ACM SIGCOMM Computer Communication Review*, 34(4):79–92, 2004.

[GRS96] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding Routing Information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, May 1996.

[GTPZ10] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @spam: the underground on 140 characters or less. In *CCS '10: Proceedings of the 17th ACM conference on Computer and communications security*, pages 27–37. ACM, 2010.

[HCJS03] F. Hernández-Campos, K. Jeffay, and F.D. Smith. Tracking the evolution of web traffic: 1995-2003. In *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*, pages 16–25. IEEE, 2003.

[imp05] The impact of AJAX on web operations, 2005. `http://www.bitcurrent.com/105/`.

[int] Wikipedia - List of countries by number of Internet users. `http://en.wikipedia.org/wiki/List_of_countries_by_number_of_Internet_users`.

[ipoa] ipoque. Internet Study 2007. `http://www.ipoque.com/news_&_events/internet_studies/internet_study_2007`.

[ipob] ipoque. Internet Study 2008/2009. `http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009`.

[IUKB+04] Mikel Izal, Guillaume Urvoy-Keller, Ernst W. Biersack, Pascal A. Felber, Al Anwar Hamra, and Luis Garces-Erice. Dissecting BitTorrent: Five Months in a Torrent's Lifetime. *PAM '04: Proceedings of the 5th Passive And Active Measurement Conference*, pages 1–11, 2004.

[JCDK01] Kirk L. Johnson, John F. Carr, Mark S. Day, and M. Frans Kaashoek. The Measured Performance of Content Distribution Networks. *Computer Communications*, 24(2):202–206, 2001.

[Kar09] Karsten Loesing. Measuring The Tor Network: Evaluation Of Client Requests to the Directories, 2009. `https://git.torproject.org/checkout/metrics/master/report/dirreq/directory-requests-2009-06-26.pdf`.

[KBJK+06] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. Towards IP geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet Measurements*, pages 71–84. ACM New York, NY, USA, 2006.

[KCF+08] Hyunchul Kim, KC Claffy, Marina Fomenkov, Dhiman Barman, Michalis Faloutsos, and KiYoung Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In *CoNEXT '08: Proceedings of the 2008 ACM CoNEXT Conference*, pages 1–12, New York, NY, USA, 2008. ACM.

[KLPM10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 591–600, New York, NY, USA, 2010. ACM.

[KNT06] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD '06: Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, 2006.

[KRP05] Thomas Karagiannis, Pablo Rodriguez, and Konstantina Papagiannaki. Should internet service providers fear peer-assisted content distribution? In *Proceedings*

of the Internet Measurement Conference 2005 on Internet Measurement Conference table of contents, pages 6–6. USENIX Association Berkeley, CA, USA, 2005.

[KW00] Balachander Krishnamurthy and Craig E. Wills. Analyzing Factors that Influence End-to-End Web Performance. *Computer Networks*, 33(1-6):17–32, 2000.

[KWZ01] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 169–182. ACM New York, NY, USA, 2001.

[LAH06] J. Leskovec, L. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *Proceedings of the 7th ACM conference on Electronic commerce (EC)*, pages 228–237, 2006.

[LG10] K. Lerman and R. Ghosh. Information contagion: n empirical study of the spread of news on digg and twitter social networks. In *Proceedings of the 3th AAAI Conference on Weblogs and Social Media (ICWSM'10)*, pages 90–97, 2010.

[LKS10] Myungjin Lee, Ramana Rao Kompella, and Sumeet Singh. Ajaxtracker: active measurement system for high-fidelity characterization of ajax applications. In *WebApps'10: Proceedings of the 2010 USENIX conference on Web application development*, pages 2–2, Berkeley, CA, USA, 2010. USENIX Association.

[LMC08] W. Li, A.W. Moore, and M. Canini. Classifying HTTP traffic in the new age. In *ACM SIGCOMM Poster Session*, 2008.

[LRW03] Nathaniel Leibowitz, Matei Ripeanu, and Adam Wierzbicki. Deconstructing The Kazaa Network. In *WIAPP '03: Proceedings of the 3rd IEEE Workshop On Internet Applications*, page 112, Washington, DC, USA, 2003. IEEE Computer Society.

[LSL+10] Fangming Liu, Ye Sun, Bo Li, Baochun Li, and Xinyan Zhang. Fs2you: Peer-assisted semipersistent online hosting at a large scale. *Parallel and Distributed Systems, IEEE Transactions on*, 21(10):1442 –1457, oct. 2010.

[Mah97] B.A. Mah. An empirical model of HTTP network traffic. In *IEEE INFOCOM*, volume 2, pages 592–600. Citeseer, 1997.

[mak] Announcement of URL shortening service available at makeashorterlink.com. `http://www.metafilter.com/8916/`.

[Mar02] Evangelos P. Markatos. Tracing a large-scale peer to peer system: an hour in the life of gnutella. In *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2002.

[MBG+08] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining Light in Dark Places: Understanding the Tor Network. In Nikita Borisov and Ian Goldberg, editors, *Proceedings of the 8th International Symposium on Privacy Enhancing Technologies*, pages 63–76, Leuven, Belgium, July 2008. Springer.

[MD05] Steven J. Murdoch and George Danezis. Low-Cost Traffic Analysis of Tor. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*. IEEE CS, May 2005.

[MMG+07] A. Mislove, M. Marcon, Krishna P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proc of the 5th ACM/USENIX Internet Measurement Conference (IMC'07)*, 2007.

[MSL+11] Martin Mulazzani, Sebastian Schrittwieser, Manuel Leithner, Markus Huber, and Edgar R. Weippl. Dark clouds on the horizon: Using cloud storage as attack vector and online slack space. In *USENIX Security*, 8 2011.

[NBVA+11] Nick Nikiforakis, Marco Balduzzi, Steven Van Acker, Wouter Joosen, and Davide Balzarotti. Exposing the lack of privacy in file hosting services. In *Proceed-*

ings of the 4th USENIX conference on Large-scale exploits and emergent threats, LEET'11, pages 1–1, Berkeley, CA, USA, 2011. USENIX Association.

[NRC08] A. Nazir, S. Raza, and C. Chuah. Unveiling facebook: a measurement study of social network based applications. In *IMC '08: Proc. of the ACM SIGCOMM conference on Internet measurement*, pages 43–56, 2008.

[NRS+02] Diego Nogueira, Leonardo Rocha, Juliano Santos, Paulo Araujo, Virgilio Almeida, and Meira Jr. Weira. A methodology for workload characterization of file-sharing peer-to-peer networks. In *Workload Characterization, 2002. WWC-5. 2002 IEEE International Workshop on*, pages 118–126, 2002.

[OR93] J. Oikarinen and D. Reed. Internet relay chat protocol. 1993.

[Or07] T. O reilly. What is web 2.0: Design patterns and business models for the next generation of software. *Communications and Strategies*, 65:17, 2007.

[PAIM08] Vasilis Pappas, Elias Athanasopoulos, Sotiris Ioannidis, and Evangelos P. Markatos. Compromising Anonymity Using Packet Spinning. In *Proceedings of the 11th Information Security Conference*, September 2008.

[PD07] Ravi S. Prasad and Constantine Dovrolis. Measuring the congestion responsiveness of internet traffic. In *PAM'07: Proceedings of the 8th international conference on Passive and active network measurement*, pages 176–185, Berlin, Heidelberg, 2007. Springer-Verlag.

[PFA+10a] Ingmar Poese, Benjamin Frank, Bernhard Ager, Georgios Smaragdakis, and Anja Feldmann. Improving content delivery using provider-aided distance information. In *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pages 22–34, New York, NY, USA, 2010. ACM.

[PFA+10b] Ingmar Poese, Benjamin Frank, Bernhard Ager, Georgios Smaragdakis, and Anja Feldmann. Improving content delivery using provider-aided distance information.

In *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pages 22–34, New York, NY, USA, 2010. ACM.

[PR85] J. Postel and J. Reynolds. File transfer protocol. 1985.

[RBC+11] Tiago Rodrigues, Fabrício Benevenuto, Meeyoung Cha, Krishna P. Gummadi, and Virgílio Almeida. On word-of-mouth based discovery of the web. In *ACM SIGCOMM 11th Internet Measurement Conference (IMC)*, 2011.

[RFI02] Matei Ripeanu, Ian Foster, and Adriana Iamnitchi. Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 2002.

[RGKW08] Charles Reis, Steven D. Gribble, Tadayoshi Kohno, and Nicholas C. Weaver. Detecting In-flight Page Changes with Web Tripwires. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 31–44, Berkeley, CA, USA, 2008. USENIX Association.

[SAAF08] Fabian Schneider, Sachin Agarwal, Tansu Alpcan, and Anja Feldmann. The new web: characterizing ajax traffic. In *PAM'08: Proceedings of the 9th international conference on Passive and active network measure ment*, pages 31–40, Berlin, Heidelberg, 2008. Springer-Verlag.

[sar02] An Analysis Of Internet Content Delivery Systems. pages 315–327, 2002.

[SB08] Robin Snader and Nikita Borisov. A Tune-up for Tor: Improving Security and Performance in the Tor Network. In *Proceedings of the Network and Distributed Security Symposium*. Internet Society, February 2008.

[SCJO01] F. Donelson Smith, Félix Hernández Campos, Kevin Jeffay, and David Ott. What tcp/ip protocol headers can tell us about the web. *SIGMETRICS Perform. Eval. Rev.*, 29(1):245–256, 2001.

[SGG03] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia Systems*, 9(2):170–184, 2003.

[SPBP06] Neil Spring, Larry Peterson, Andy Bavier, and Vivek Pai. Using PlanetLab For Network Research: Myths, Realities, And Best Practices. *ACM SIGOPS Operating Systems Review*, 40(1):17–24, 2006.

[SW02] Subhabrata Sen and Jia Wang. Analyzing peer-to-peer traffic across large networks. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 137–150, New York, NY, USA, 2002. ACM.

[The] The Tor Project. TorCtl. `https://svn.torproject.org/cgi-bin/viewvc.cgi/torctl/`.

[tin] TinyURL.com. `http://tinyurl.com/`.

[twe] TweetDeck. `http://www.tweetdeck.com/`.

[twia] Twitter Rate Limit. `http://apiwiki.twitter.com/Rate-limiting`.

[twib] Twitter Search. `http://search.twitter.com/`.

[twic] TwitterFeed. `http://twitterfeed.com/`.

[WGM+09] Helen J. Wang, Chris Grier, Alexander Moshchuk, Samuel T. King, Piali Choudhury, and Herman Venter. The multi-principal os construction of the gazelle web browser. In *SSYM'09: Proceedings of the 18th conference on USENIX security symposium*, pages 417–432, Berkeley, CA, USA, 2009. USENIX Association.

[WHMW11] Shaomei Wu, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 705–714, New York, NY, USA, 2011. ACM.