INVESTIGATING THE MUTUAL IMPACT OF THE P2P OVERLAY AND THE AS-LEVEL UNDERLAY

by

HASSAN RASTI EKBATANI

A DISSERTATION

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

December 2012

DISSERTATION APPROVAL PAGE

Student: Hassan Rasti Ekbatani

Title: Investigating the Mutual Impact of the P2P Overlay and the AS-level Underlay

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Computer and Information Science by:

Prof. Reza Rejaie	Chair
Prof. Virginia Lo	Inside Member
Prof. Arthur Farley	Inside Member
Prof. David Levin	Outside Member

and

Prof. Kimberly Andrews Espy	Vice President for Research & Innovation/ Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded December 2012

DISSERTATION ABSTRACT

Hassan Rasti Ekbatani

Doctor of Philosophy

Department of Computer and Information Science

December 2012

Title: Investigating the Mutual Impact of the P2P Overlay and the AS-level Underlay

During the past decade, the Internet has witnessed a dramatic increase in the popularity of Peer-to-Peer (P2P) applications. This has caused a significant growth in the volume of P2P traffic. This trend has been particularly alarming for the Internet Service Providers (ISPs) that need to cope with the associated cost but have limited control in routing or managing P2P traffic. To alleviate this problem, researchers have proposed mechanisms to reduce the volume of external P2P traffic for individual ISPs. However, prior studies have not examined the global effect of P2P applications on the entire network, namely the traffic that a P2P application imposes on individual underlying Autonomous Systems (ASs). Such a global view is particularly important because of the large number of geographically scattered peers in P2P applications.

This dissertation examines the global effect of P2P applications on the underlying AS-level Internet. Toward this end, first we leverage a large number of complete overlay snapshots from a large-scale P2P application, namely Gnutella, to characterize the connectivity and evolution of its overlay structure. We also conduct a case study on the performance of BitTorrent and its correlation with peer- and group-level properties. Second, we present and evaluate Respondent-driven sampling as a promising technique to collect unbiased samples for characterizing peer properties in large-scale P2P overlays without requiring the overlay's complete snapshot. Third, we propose a new technique leveraging the geographical location of peers in an AS to determine its geographical footprint and identify the cities where its Points-of-Presence (PoPs) are likely to be located. Fourth, we present a new methodology to characterize the effect of a given P2P overlay on the underlying ASs. Our approach relies on the large scale simulation of BGP routing over the AS-level snapshots of the Internet to identify the imposed load on each transit AS. Using our methodology, we characterize the impact of Gnutella overlay on the AS-level underlay over a 4-year period. Our investigation provides valuable insights on the global impact of large scale P2P overlay on individual ASs.

This dissertation includes my previously published and co-authored material.

CURRICULUM VITAE

NAME OF AUTHOR: Hassan Rasti Ekbatani

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, Oregon, USA
Sharif University of Technology, Tehran, Iran

DEGREES AWARDED:

Doctor of Philosophy in Computer and Information Science, 2012, University of Oregon

Bachelor of Science in Electrical Engineering, 2001, Sharif University of Technology

AREAS OF SPECIAL INTEREST:

Network Measurement, P2P Applications, Computer Networks

PROFESSIONAL EXPERIENCE:

Graduate Research Fellow, Department of Computer and Information Science, University of Oregon, Eugene, Oregon, 2004 - 2011

Research Fellow, Institute for Pure and Applied Mathematics, UCLA, Los Angeles, California, 2008

Summer Intern, Bell Labs, Alcatel-Lucent Inc., Holmdel, New Jersey, 2008

Network Engineer, Iranian Research and Academic Network, Tehran, Iran, 2002-2004

Network Engineer, Computing Center, Sharif University of Technology, Tehran, Iran, 2000-2003

GRANTS, AWARDS AND HONORS:

UO Annual Programming Competitions; First place in the graduate division, University of Oregon, Eugene, Oregon, 2009 and 2010

Juilfs Scholarship Award; Department of Computer Science, University of Oregon, Eugene, Oregon, 2008

Membership to Upsilon Pi Epsilon International Honor Society for the Computing and Information Disciplines, University of Oregon, Eugene, Oregon, 2008

Travel Grant to Attend CoNext 2007 Conference, New York, New York, 2007

Travel Grant to Attend ICNP 2006 Conference, Santa Barbara, California, 2006

Silver Medal of the National Informatics Olympiad, Mashhad, Iran, 1994

PUBLICATIONS:

A. H. Rasti, N. Magharei, R. Rejaie, and W. Willinger. Eyeball ASes: From Geography to Connectivity. Proceedings of *ACM SIGCOMM Internet Measurement Conference*, Melbourne, Australia, Nov. 2010.

A. H. Rasti, R. Rejaie, and W. Willinger. Characterizing the Global Impact of P2P Overlays on the AS-Level Underlay. Proceedings of *Passive and Active Measurement Conference*, Zurich, Switzerland, Apr. 2010.

A. H. Rasti, M. Torkjazi, R. Rejaie, N. Duffield, W. Willinger, and D. Stutzbach. Respondent-driven Sampling for Characterizing Unstructured Overlays. Proceedings of *IEEE INFOCOM Mini-conference*, Rio de Janeiro, Brazil, Apr. 2009.

A. H. Rasti and R. Rejaie. Understanding Peer-Level Performance in BitTorrent: A Measurement Study. Proceedings of *International Conference on Computer Communications and Networks*, Honolulu, Hawaii, Aug. 2007.

A. H. Rasti, D. Stutzbach, and R. Rejaie. On the Long-term Evolution of the Two-Tier Gnutella Overlay. Proceedings of *IEEE Global Internet Symposium*, Barcelona, Spain, Apr. 2006.

ACKNOWLEDGEMENTS

My highest level of gratitude goes to God almighty for his guidance and for putting all of the following wonderful people in the way of my life. Many thanks to my advisor Prof. Reza Rejaie for believing in me and for all of his time and effort during my PhD education years that made this dissertation possible. His leadership and strict training were extremely valuable in helping me move forward. I am grateful to my committee members for their valuable feedback, especially Prof. Virginia Lo for her encouragements during the hard times and for her invaluable assistance during the final editing. I would like to thank all the people at the Department of Computer and Information Science for what I have learned from them and for their friendship, in particular, Star Holmberg who was always ready to go beyond line of duty and never hesitated to provide help. Additionally, I would like to thank my co-authors Dr. Daniel Stutzbach, Dr. Nick Duffield, and Mojtaba Torkjazi for their assistance, and my special thanks to Dr. Walter Willinger , who is never short of new ideas, for the opportunities he made possible. I would like to acknowledge the Institute for Pure and Applied Mathematics at UCLA for offering me a fellowship. I am grateful to Dr. Markus Hofmann and his excellent team at Bell Labs from whom I learned a lot. I would like to thank Masoud Valafar, Ghulam Memon, Kaveh Kazemi, and Peter Boothe for their insightful discussions, technical assistance and their friendship. Thanks to Amin Khosrowshahi, Bamdad Ehterami and Reza Motamedi for their help with setting up my remote defense session. I am very grateful to my parents for all their support throughout my education. Their encouragements has been a strong driving force.

Finally and most importantly, I am extremely thankful to my best friend, colleague and wife, Dr. Nazanin Magharei whose love and support have been the key factors in my every achievement. Nazanin has been standing by me through all the difficulties during my graduate studies and has been sharing all the hardship. Her presence has enabled me to go through this path.

To my better half, Nazanin.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

Nearly 25 years after introduction of the first TCP/IP-based wide area network, the Internet has become a complex phenomenon and a vital part of human life. It has grown enormously in size and structure, while exhibiting rapid and dynamic changes in behavior. Because the Internet plays such an important role in society, it is imperative that its characteristics are thoroughly studied and understood. However, the decentralized and distributed nature of the Internet makes this task extremely challenging. Efforts to build a complete network topology map of the Internet, for example, have focused primarily on building representative models of the Internet topology due to the inherent difficulty of building an accurate map of such a large scale and complex phenomenon. Adding to this challenge is the fact that the Internet serves as the infrastructure for a diverse group of players, each seeking their own goals. While Internet Service Providers (ISPs), network providers and content providers try to maximize their specific benefits, governments try to enforce traditional laws in this new domain.

In parallel with the development of traditional client/server-based network applications, a group of peer-to-peer applications emerged, in which participating users (peers) connect together, forming a fully-decentralized overlay network to assist each other towards the application goal. The P2P concept provides a self-scalable, low-cost, user-centric structure for a variety of applications, primarily content distribution. P2P applications quickly became popular enough to account for up to 70% of Internet traffic according to some reports. Their popularity and unique features makes the study of P2P applications an important research subject.

However, popular P2P applications such as BitTorrent and Gnutella form dynamic and large scale distributed systems for which capturing accurate and complete pictures is infeasible. Thus, researchers have tried to provide representative snapshots to study the features and shortcomings of various important P2P applications.

Clearly, these P2P overlay-based applications have a huge impact on the underlying network with serious economic and social consequences. For instance, with the widespread deployment of peer-to-peer file sharing applications, some ISPs tried to block or limit the traffic associated with P2P applications while the users and application developers have tried to evade these limitations. The high volume of P2P traffic is unprecedented, and the pattern of P2P traffic is dramatically different from what the ISPs expect and are provisioned for. Yet to date, very little research has focused on understanding how the behavior of P2P applications affects the Internet. Our work seeks to fill this gap.

1.1. Overview of the Problem

The key question addressed by this dissertation is the following: "What is the impact of P2P applications on the underlying network?" Answering this question involves the following *challenges*: (i) measurement and characterization of P2P/overlay applications, (ii) capturing and characterization of the Internets autonomous systems (AS-level) topology, and (iii) characterization of the impact of P2P applications on the underlying AS-level network.

1.1.1. Measurement and Characterization of P2P Applications

Popular P2P applications are dynamic large scale distributed systems with global footprints. In the case of pure P2P systems, there is no entity that holds all the information about participating peers and their connections. In some cases, there are bootstrapping servers with partial information about participating peers; however, they typically do not share such information with the academic research community and they often do not have connectivity information. The dynamic nature of these systems is another challenge for the researcher as it often results in distorted pictures of the system.

1.1.2. Characterization of the Internet's AS-level Topology

The complex challenge we face in understanding the Internets AS-level topology is due to many aspects of this network:

- The Internet is a huge network with more than 30,000 ASs, hundreds of thousands of routers and billions of nodes. Mapping a network of this scale is hard, even if there was no limitation on data gathering.

- The Internet is a decentralized network of networks. No central entity has complete information about all the networks. Each AS makes internal decisions independently and each pair of ASs may establish links for private or public traffic exchange. Commonly used techniques for gathering Internet topology information (*e.g.*, traceroute-based and BGP-based) have well-known shortcomings and the resulting maps are known to be incomplete.

- Business, political or security motivations impact ASs willingness to share information about their internal topology, user population, business agreements, costs, routing preferences, etc. Therefore direct inquiry (including the companies websites) may be of limited use.

– A limited number of AS-level paths can be directly extracted from the archived BGP tables. Besides this subset, finding paths among pairs of ASs is a challenging task that requires information about the connectivity among ASs as well as routing preferences at each ISP. Although there are available techniques for inferring the types of relationships among connected ASs, they have limited accuracy.

1.1.3. Characterizing the Impact of the Overlay on the Underlay

How then can we hope to understand the impact of large, fully-decentralized, highly dynamic P2P applications on the even more vast, complex, and dynamic AS-level network? How does the P2P overlay network map to the underlying AS-level overlay network? Using archived BGP tables, we could try to map each IP address in the overlay to their respective AS. However for many ASs, different geographical parts of the AS may have different routing preferences. In these cases, pinpointing the actual AS-level path from one IP host to another becomes extremely challenging. This problem also holds in the general case of multi-homed ASs (those with multiple providers) that perform load balancing among their providers. Clearly, these questions require new advances in networking research to address a range of challenging problems.

1.2. Our Approach

Our approach to characterizing the impact of a P2P/overlay application on the underlying AS-level network breaks the problem down into the three steps discussed above. We discuss our work in more detail below.

1.2.1. Measurement and Characterization of P2P Applications

As described earlier, P2P applications form large-scale dynamic systems. Capturing an instant image of the system is not possible due to decentralization and lack of central control. We have developed new efficient and effective crawlers to capture snapshots of large-scale P2P applications to study the topology, application behavior, as well as user behavior. As a case study, we examine Gnutella file sharing application. We gathered full snapshots of Gnutella over the course of two years and presented the long-term trends and evolution of this system. We also performed a measurement study of peer performance in BitTorrent. We used tracker logs for popular torrents from which we extract each peers download and upload rates during their session times and examine the root causes for the observed performance by individual peers.

Depending on the size of the network and crawling speed, the picture provided by a crawler can be distorted due to the time factor involved in data gathering (the system changes while the picture is being taken). Thus, we use sampling techniques to capture a representative subset of the system in a short time. In particular we implement a special version of respondent-driven sampling for this purpose.

1.2.2. Characterization of the Internets AS-level Topology

When we study the Internet as a network of networks, we can use the AS-level abstraction. In an AS-level topology, nodes represent ASs and edges represent connectivity and peering relationships among ASs. Current methods for capturing an AS-level map of the Internet have known limitations. We use a combination of available data sources and techniques including archived BGP tables and IXP

(Internet exchange point) membership information to build an AS-level connectivity and relationship map.

In order to find AS-level paths, we translate the inter-AS relationships into common routing policies and simulate BGP routing on a full AS-level network. Although the accuracy of the resulting paths may be questionable, we argue that they can be used to represent the actual paths, because; (i) the actual paths often change due to load balancing, and (ii) the characteristics (e.g., topological and hierarchical) are representative of the actual paths.

The available methods for capturing AS-level topology do not provide a complete picture. Both BGP-based and traceroute-based methods have limited views of the inter-AS connectivity. Although they provide a rather complete picture of the core of the network, they have limited vision on the edge.

We propose a complementary method in which we characterize the geographical footprints of eyeball ASs (end-user ISPs). In this method, we use snapshots of popular P2P applications as user IP address pools. For each eyeball AS, we extract their part of user IP pool and map the IP addresses into geographical locations using commercially available tools. Next, a two dimensional kernel method is used to derive a geographical user-density function for each AS. The resulting function is used to estimate the PoPs (Points of Presence) for each AS, as well as to visualize the geographical footprint of the AS.

We study the underlying factors behind the observed AS-level topology. We explore the reasons behind peering relationships among ASs and find out which factors are more important in ISP decisions on which provider to choose, which peering connection to establish and which IXP to attend to. We consider two groups of goals: (i) business goals: *i.e.*, each AS tries to minimize costs and maximize benefits, and

(ii) geographical feasibility: *i.e.*, ASs that are not in the geographical vicinity of each other are less likely to establish direct peering.

1.2.3. Characterizing the Impact of the Overlay on the Underlay

The large and growing traffic associated with P2P applications and the concern among ISPs who need to carry this traffic have led researchers to focus on the idea of making P2P traffic less network-costly. However, the global impact of P2P applications on the underlying network is not well understood. For example, it is not known what portion of the P2P traffic is local (in terms of AS, country or continent), how far the traffic has to go up in the AS hierarchy, or how the answer to these questions differ across different P2P applications. We tackle this problem by using the Gnutella P2P application as a sample case, using the datasets and tools we developed. We aggregate peers based on which AS they come from and then we assume a simplistic traffic model on the overlay. We form a global AS-level traffic demand matrix in which the traffic demand between each pair of ASs is presented. We also present the AS-level paths in a sparse binary matrix format and then we derive a traffic matrix showing the amount of traffic on the links between connected ASs. By plugging in any desirable P2P overlay map and any traffic model, this method provides the traffic matrix on the underlying AS-level network. By examining the resulting traffic matrix we can characterize the impact of any P2P overlay application.

1.3. Roadmap

The rest of this dissertation is organized as follows. In Chapter II, we present the background and the related work of this research. Chapter III presents our measurement study on Gnutella P2P application. In Chapter IV, we propose a novel sampling technique for capturing peer characteristics in large scale overlay networks. This technique complements the method used in Chapter III (capturing full snapshots) by enabling the researcher to capture peer properties of a very large scale P2P application in a short time. In Chapter V, we present our case study on P2P performance evaluation focusing on BitTorrent. This chapter provides another perspective of P2P applications by focusing on the user experience rather than overlay characteristics. Also, we showcase another technique for characterizing P2P applications in which we do not perform any active or passive measurement. Instead, we perform a sophisticated set of analyses on the log files provided by the P2P bootstrapping hosts. Next, we turn our attention to the AS-level underlay. Chapter VI presents our study on geographical mapping of ASs. In this chapter, we use the datasets gathered in the study presented in Chapter III in addition to other P2P snapshots we gathered later and propose a novel method for geographical mapping of ASs. Chapter VII presents our work on the impact of P2P overlay on the AS-level underlay. In this chapter, we use the technique and tool described in Chapter III for gathering a large collection of P2P snapshots and showcase our novel method for assessing the traffic imposed by a P2P application on the underlying network of ASs. Finally, we present concluding remarks, summary of contributions and future directions in Chapter VIII.

Chapters III,IV,V,VI,VII of this dissertation are based heavily on my published papers with co-authors [102, 105, 103, 106, 104]. In all of the works, the experimental work is entirely mine, with my co-authors contributing technical guidance, editorial assistance, and portions of writing.

CHAPTER II

BACKGROUND AND RELATED WORK

The Internet has evolved greatly since its first days in different aspects. The network infrastructure has grown from a few academic and research institutions to a huge global network with nodes in almost every home. In the meantime, a new class of applications have been designed and widely used over the Internet for a wide variety of functions; peer-to-peer (P2P) applications. In P2P applications, participating peers form overlays through which they exchange data. The load imposed by the P2P applications on the network has raised concerns in ISPs due to its high volume and different pattern compared to traditional client-server applications. These issues have motivated three areas of research : *(i)* Internet topology, *(ii)* design and characterization of P2P applications, and *(iii)* studying the mutual impacts between the P2P applications and the underlying network. In this chapter, we survey the research works published in these areas in order to locate any open issues and problems. Below, we present an overview of these areas.

2.1. Overview

In this section, we present an overview of Chapter II by providing a high-level categorization of the previously published research works related to this dissertation.

2.1.1. Internet Topology Characterization

In this area, the researchers study the Internet connectivity graphs in order to learn about the structure of the Internet and how it is evolving. Such information is critical for Internet researchers as it provides knowledge about potential features and shortcomings that may result from certain connectivity structure. For instance, some studies (*e.g.*, [3]) have claimed that the Internet has a *scale-free* structure and therefore its connectivity is dependent on a small number of very high degree nodes (hubs) and concluded that the Internet is vulnerable to targeted attacks on these hubs.

The Internet topology is often studied at two different abstraction levels: *(i) Router-level topology* describes the connectivity graph of the routers that interconnect the Internet, while in *(ii) AS-level topology* the connectivity of autonomous systems (*i.e.*, networks with an independent management) is the subject of study. Since the expansion of the Internet to a global network, no complete topology of the Internet has been presented to date and such a task still remains infeasible to do due to the distributed nature of the Internet. Despite this fact and other challenges, a significant number of research studies have been working on capturing and characterizing the Internet topology at both AS- and router-level using innovative techniques that we will discuss in Section 2.3..

2.1.2. P2P Application Design and Characterization

The attractive features of the P2P network application model has encouraged application developers to employ the P2P model in a variety of applications. Specifically, in the area of content delivery and sharing, P2P applications have been mostly successful and popular. Nevertheless, designing an efficient, reliable and high performance P2P application can be very challenging. In such systems dynamics of peer participation, heterogeneity of the peers in terms of available resources and bandwidth along with some other issues are the main challenges that an application designer has to overcome. We will discuss some of the research works in this area in Section 2.2.1..

Once a P2P application is widely adopted by the Internet users, there are still many questions that need to be answered about it. Due to the distributed and nature of these applications, there is often no central monitoring or controlling entity and therefore one cannot answer questions on issues such as the performance and efficiency of the working system without a thorough network measurement. Also, the researchers are often interested in studying large P2P overlay networks as samples of complex networks in order to discover their features and shortcomings. In Section 2.2.2., we discuss some of the mostly cited works in the area of P2P measurement and characterization.

2.1.3. Overlay-Underlay Interaction

We mentioned before that the participating peers in a P2P application form an overlay. An overlay is a virtual data communication network that is built over a real network (Internet) actually responsible to carry the data packets. In recent years, the traffic imposed by the P2P overlays has raised concerns in many ISPs urging them to limit or control this traffic. In the area of overlay-underlay interaction, the researchers discuss the following issues: *(i)* The impact (load) of the P2P overlays on the network, *(ii)* ISP efforts to limit the impact and the reaction of P2P applications, *(iii)* ISP-friendly P2P applications, and *(iv)* ISP-P2P cooperation.

The increasing popularity of the peer-to-peer applications has caused the traffic of such systems to become an issue for the ISPs. On one hand, the P2P model is attractive to the content providers because it empowers them to feed more users with little investment. For instance, NBC has re-branded a P2P streaming and file sharing platform, called Pando, for high definition rebroadcasting of their shows over the Internet. On the other hand, many ISPs have raised concerns about both level and pattern of the traffic caused by P2P applications. Furthermore, some ISPs have incorporated mechanisms to detect and limit the amount of traffic associated with certain P2P applications. In the summer of 2008, the Federal Communications Commission (FCC) issued a ruling against Comcast on "discrimination among applications" and ordering them to stop such practices. The ruling was based on a complaint accusing Comcast of blocking P2P traffic. This was after other attempts by P2P applications to make P2P traffic harder to detect and control by the ISPs (*e.g.*, encryption).

2.1.4. Grounds for ISP Concerns

Earlier we mentioned that with the growth of P2P traffic, many ISPs became concerned and took actions to limit or block P2P traffic. Here we try to answer the following question: *Why are the ISPs concerned about P2P traffic ?* There are two important differences between the P2P applications and common client-server applications; *(i)* In most of the traditional client-server applications (*e.g.*, WWW), the uplink traffic of the users is relatively small. However, in P2P applications, participating peers may generate as much upload traffic as they download. This results in a significant increase in the amount of upload traffic that the ISPs have to transmit. *(ii)* In most traditional applications, the traffic flow has a temporal dependence with the human interaction. For example, when the user clicks on a WWW link, the client starts to download the targeted page or file and the flow stops as soon as the download is complete which normally takes between a few seconds to a few minutes for larger files that are requested less frequently. In contrast, in P2P applications, although the traffic flow starts with user interaction, it will often continue much longer without any user action. For instance, in BitTorrent (a popular peer-to-peer file distribution application) the network link is often utilized in both outbound and inbound directions during the downloading time. Even once the download is complete, the client automatically continues providing content to other participants until stopped by the user effectively keeping the uplink busy even after the download is complete. The uplink traffic is often sent to peers in other ISPs and therefore increases the load on the inter-ISP links. Therefore, the advent of P2P applications increases ISPs' costs by *(i)* increasing the ISPs' uplink traffic for the same volume of download *(ii)* changing the user traffic pattern from bursty (short flows of traffic that are originated by user interactions) to steady (continuous flow even without user interaction). Assuming fixed amount of download per user, increased uplink traffic often means that the ISPs should purchase more bandwidth for the same number of users. Also, bursty traffic pattern which was the dominant user utilization pattern, allowed a much larger provisioning ratio for the ISPs (the short flows by different users occur in different times and therefore the momentary load of the ISP is small) compared to a steady pattern, effectively forcing ISPs to purchase larger bandwidth for a fixed number of users.

The problem of P2P traffic for ISPs has motivated several sets of research projects. Some have proposed methods to make P2P applications "ISP-friendly" mainly by localizing their traffic within ISPs. Although localization can reduce ISP load for certain scenarios without degrading the application's performance, in many cases localization may limit the performance of P2P applications, mainly by making the groups of peers that can help each other smaller. Such limitations suggest that localization is not enough and some other mechanisms need to be used to differentiate between external peers.

Recently, there has been multiple research works suggesting cooperation between peer-to-peer applications and the ISPs. In summary, within such cooperative methods, ISPs help peer-to-peer applications select neighbors in order to minimize the load on the ISP's costly links. The ISP uses its information about the topology, link costs and utilization in order to adjust the amount of P2P traffic on its own external links.

2.1.5. Roadmap

In this chapter, we survey and categorize the research works in the three areas mentioned above. We aim to understand the mutual effects of the P2P overlays and the Internet underlay. However, in order to characterize such effects, we first need to understand the characteristics of the P2P overlays and the underlying network. Toward this end, in Section 2.2., we survey research studies on the design and characterization of P2P applications. Section 2.2.1. categorizes P2P overlays according to the function, structure and shape of the overlay. Section 2.2.2. surveys and groups research works in the area of characterizing P2P overlays through measurement, modeling and simulation. In Section 2.3. we survey some outstanding research studies on the Internet topology. In Section 2.3.1., we discuss research studies characterizing the AS-level topology of the Internet and categorize their data gathering and characterization methods. Section 2.3.2. surveys the studies on router-level topology of the Internet and categorizes their data sources and characterizations.

Section 2.4., we focus on the mutual impacts of the P2P overlays and the underlying network. Section 2.4.1. surveys research works on the impact of P2P overlays on the network. In Section 2.4.2. discusses the actions made by the ISPs to manage the P2P traffic and why they are not acceptable by the network community. In Section 2.4.3., we summarize research works that try to form ISP-friendly overlays and in Section 2.4.4., we survey the recent works based on the cooperation between the ISP and the P2P applications. Finally, Section 2.5. concludes the chapter by reviewing the main challenges and shortcomings.

2.2. Overlay Networks

P2P applications are used to provide a variety of network services in a decentralized fashion. Such systems are: *(i) robust*, since they do not have a single point of failure; *(ii) scalable*, as each user adds resources to the system, and capable of functioning at *(iii) very low cost*. The collection of participating peers in a P2P network form a *P2P overlay* which is a virtual network over which the peers exchange data. In most of today's P2P systems, the overlay networks are formed without considering the underlying network. For instance, in a random overlay network two peers that are in the same physical network have only a small chance of getting connected to each other while each may have neighbors from across the globe. Besides random overlays, in another group of P2P applications the overlay construction may have particular goals. For instance, in a gaming overlay, it is fair to assume that minimizing delay (between interacting peers) should be the goal of overlay construction while in streaming, maximum bandwidth from the source might be as important. In this class of P2P overlays, the goals of minimizing delay and bandwidth may indirectly cause overlay connections to become more localized.

Finally, a few recently proposed P2P overlays explicitly follow the goal of locality-awareness or network-awareness either with or without the support from the underlying network [1]. Considering the popularity of the P2P applications and the load they impose on the underlying network, it is important to study different types of P2P applications as well as the research works aiming at characterizing P2P overlays.

The impact of a P2P overlay on the underlying network depends on: *(i)* overlay connectivity structure, *(ii) traffic generation pattern* and *(iii)* packet forwarding and routing mechanism in the overlay. In order to study this imapct, we need to learn about the structure, packet generation and data paths in the overlays. The P2P applications are used for a variety of functions and their respective overlay networks have different shapes, structures and characteristics according to the functionality they are designed for. In Section 2.2.1., we categorize most well-known P2P overlay networks in research and user communities according to the overlay's functionality, structure and shapes and compare the subgroups, accordingly. In Section 2.2.2. we overview a number of research works on characterizing P2P overlays while categorizing them according to their approaches.

2.2.1. Categorizing P2P Overlays

P2P applications can be categorized from numerous aspects. In this report, we focus on the overlays and therefore we categorize P2P applications with this focus. Although overlays may be used for a variety of purposes, generally one overlay is constructed and used for a single functionality. In some P2P applications, multiple overlays are formed and used for multiple functionalities. This is because the structure, shape, and characteristics of the overlay depends on its functionality.

The two main classes of overlay functionalities are *(i) Signaling and control* and *(ii) Content delivery.* In content delivery overlays, large amounts of data

[1] We will discuss this issue in detail in Section 2.4.

are transferred through the overlay to reach interested peers while in signaling overlays only queries and responses that are often short are transmitted through the overlay. We discuss and further subgroup signaling and content delivery overlays in Sections 2.2.1.1. and 2.2.1.2., respectively.

2.2.1.1. Signaling and Control Overlays

In a variety of P2P applications, overlays are used for maintaining membership and exchanging queries and responses. In this class of overlays, the main goal of the overlay construction are *reachability* and *resiliency* and therefore the overlays are often richly connected. The following are some examples of the signaling overlays categorized by the functions.

2.2.1.1.1. Categorizing Signaling Overlays Based on Functionality

– **Searching** is an important problem in file sharing applications. In these applications each user shares a number of files with other users and is interested in finding and downloading other files shared by other users. In order to avoid single point of failure issue associated with a central indexing server (*e.g.*, Napster), a *decentralized search* mechanism is used by some file sharing applications such as Gnutella. In Gnutella, participating peers form an overlay to handle the decentralized search functionality. Peers send their search queries to their neighbors and each peer checks the query against their own shared files. If they have a matching file, they will send back a positive response, otherwise they relay the query to their neighbors. Although simple, searching over a large scale flat overlay may become quite inefficient. In Semantic Small World [77], peers form a highly clustered overlay. The clusters are based on the semantics of the content shared by peers. Taking advantage of similar interests by groups of people, the semantic based clustering makes searching much more efficient in SSW.

- **Store and lookup services** are handled by a group of popular P2P applications called *Distributed Hash Tables* (DHT). A DHT is responsible for distributed storage of key-value pairs, similar to a local hash table. In DHTs each peer is assigned with an ID and is responsible for a part of the hash space according to the assigned ID. Each peer maintains a *routing table* consisting of a set of links to other peers that are its neighbors. Together these links form the overlay network. A node picks its neighbors according to a certain structure that is the main difference between different DHTs and is often referred to as the DHT's topology. Commonly, the routing table size and the routing algorithm complexity in DHTs are $O(log(n))$ where n denotes the number of participants. In CAN[108], peers form an overlay over a virtual multi-dimensional Cartesian coordinate space. This d-dimensional coordinate space is a virtual logical address, completely independent of the physical location and physical connectivity of the nodes. In Chord[126], node keys are arranged on a circle. Each peer's routing table includes its successor and predecessor which are the next and the previous node on the circle, respectively. Each peer is responsible for the ID space contained between that peer and its successor. In addition to the successor and predecessor the routing tables also include a few shortcuts to other locations in the circle for the sake of faster routing.

  Other well-known proposed DHTs are Pastry[113], Tapestry[141] both using circular ID spaces and Kademlia[87] uses the XOR metric to calculate the binary distance between two peers, in order to determine neighboring and routing information. Due to the efficient decentralized store/lookup service they offer, DHTs are widely used in different applications for indexing and state keeping. For instance, in Vuze, a popular BitTorrent client, a DHT is formed to act instead of a BitTorrent tracker, in case it becomes unavailable. In Freenet[34] a DHT-like overlay is formed for anonymized distribution of data to protect freedom of speech. The protocol design ensures anonymity of the publisher and downloaders of the data.

2.2.1.1.2. Categorizing signaling overlays based on structure

Signaling overlays are generally divided into two groups based on the their structure. Below we compare and contrast the two groups with examples.

– **Unstructured overlays:** In this group of overlays, peers connect to each other in an arbitrary fashion. Each peer can individually select its own neighbors after a *peer discovery* phase in which peers acquire information about other participating peers. The resulting overlay topology is often close to a random graph, and thereby, highly *resilient* to *churn* (*i.e.*, dynamics of peer participation).

  In Gnutella, peers upon joining follow a peer discovery mechanism and learn about a number of other participating peers. Among those peers, they randomly select a subset and try connecting to them and continue until a predefined number of neighbors is reached. However, our study on Gnutella presented in Chapter III shows that there is a certain level of connectivity preference towards geographically close peers that may allow one to argue that Gnutella overlay is not purely random.

Although unstructured overlays are easy to build and maintain, their *performance* and *efficiency* are often points of concern. Searching for popular content in an unstructured overlay is often easy and fast, while the search performance for unpopular content is lower. This is because the query should eventually reach all the participating peers to ensure that a rare content can be found. There is also a trade-off between efficiency and performance of searching that a P2P application can control by adjusting the forwarding range of each query.

Although signaling overlays usually do not carry heavy traffic, high packet rate may become an issue for large flat overlays. To alleviate this problem multiple techniques have been used including the two-tier topology in modern Gnutella.

– **Structured overlays:** In structured overlays, also known as distributed hash tables (DHT), globally consistent protocols are used for neighbor selection and query routing in order to ensure efficient routing and resolution of queries. CAN[108], Chord[126], Pastry[113], Tapestry[141] and Kademlia[87] are the most well-known DHTs and we briefly discussed them earlier.

Structured overlays can offer high levels of performance and efficiency. Most operations, such as joining the overlay and looking up a key value are performed in $O(log(n))$ where n denotes the overlay size. However, maintaining the overlay in presence of churn is often quite costly. When a peer leaves the DHT, its responsibility should be transferred to other peers. Also when a new peer joins the system, it should find its place and often load the keys previously stored in its responsibility zone from other peers. Additionally, most DHTs require periodic maintenance to keep the space allocation balanced and their routing tables up-to-date.

Although most structured overlays are used for store-lookup services, there are exceptions such as Freenet [34] in which published files by the users are stored in the overlay in order to provide an anonymous and non-traceable file sharing environment.

2.2.1.2. Content Delivery Overlays

In this class of overlays, participating peers assist each other in downloading the content by contributing their upload *bandwidth.* The content is either a file or a stream which all peers are interested in receiving. The content is often broken into chunks and transmitted through the overlay and relayed by each peer to reach all other peers. In the traditional client-server content distribution, the server needs to have a large bandwidth as well as other resources in order to serve all the clients. However, in P2P content delivery, the source will only need to upload the content a small number of times (ideally once) and then the peers will download the content from each other. Thereby, content delivery overlays provide a *scalable*, *resilient* and *low cost* method for distributing large files and streams and therefore have become very popular.

In this section we divide the content delivery overlays from 3 aspects: *(i) content-type*, *(ii) overlay shape*, and *(iii) content delivery mechanism.*

2.2.1.2.1. Categorizing Content Delivery Overlays Based on the Content Type

The content distributed through the overlay may be a *file* or a *stream.*

- **File distribution overlays:** In these P2P applications, a file or a set of files are shared by a source among the participating peers. Although the downloading peer's goal is to complete the download as fast as possible, there is no hard timing constraints and therefore this class of content is also referred to as *elastic* content. BitTorrent [36] is the most popular P2P file distribution application. In BitTorrent, users interested in downloading the same file or set of files form a dynamic content delivery overlay. The files are divided into small blocks. Each peer receives the list of blocks available in its neighboring peers and subsequently sends requests for the blocks that it needs. While the *tit-for-tat* mechanism ensures bandwidth contribution by all peers, the *rarest-first* policy used by each peer for selecting which block to download, facilitates diffusion of all the blocks across the overlay. The content delivery method used by BitTorrent in which the data is broken down to small blocks which are undeterministically distributed in an overlay is also called *swarming.*

– **P2P streaming overlays:** In this class of P2P applications, multimedia streams are shared among interested users. In comparison to file distribution, streams are more challenging to distribute through an overlay due to strict timing requirements. In particular, each block of the stream will be useful at each peer only if it arrives before its playout time (non-elastic). Also, a sustained average delivery rate, equal to the stream bandwidth is necessary to each peer in order to ensure uninterrupted playback of the live stream.

  These overlays are used in delivering two types of streams. While some of the streaming overlays such as PRIME[82] and Coolstreaming [140] focus on delivering *live* audio-video streams, another group of P2P streaming applications such as Pando provide streaming of pre-recorded media often with VCR functionality. In pre-recorded streaming, a longer portion of the stream can be buffered to prevent interruptions during the playback, making the timing requirements looser. In live streaming, the amount of acceptable buffering is usually shorter. On the other hand, with pre-recorded streaming, participating peers play different parts of the stream at the same time and therefore the possibility of mutual uploading between two peers is very limited.

2.2.1.2.2. Categorizing Content Delivery Overlays Based on the Overlay Shape

Content delivery overlays are designed in one of the following shapes: (i) tree, (ii) multiple-tree, and (iii) mesh.

- **Tree:** In a *tree-based* overlay, peers form a single source-rooted tree and the content is distributed to all peers along the tree. In tree-based overlays such as Narada[33], each peer has only one parent from which all the content is downloaded. Tree-based overlays are simple to build, however they often suffer from multiple shortcomings including limited robustness and stability in presence of churn and limited scalability in terms of control overhead and latency.

- **Multiple-trees:** In more recent proposed works such as CoopNet[96] and Splitstream[23], *multiple trees* are built for content delivery overlays. In multiple-tree based overlays, the content (usually a stream) is divided to multiple parts and each part is delivered through one tree. This mechanism has three main advantages over a single tree approach: (i) In a single-tree overlay, the leaves do not contribute any bandwidth to the system while in multiple tree, each leave in one tree may have children in other trees. (ii) In a multiple-tree overlay, each peer's departure will disrupt receiving the content for all its descendents while in a multiple-tree overlay, each peer concurrently receives content from multiple parents and a temporary disconnection from one tree will only limit the rate or quality of the content. (iii) Peer heterogeneity can be supported in multiple-tree approach by joining a number of trees proportional to the peer bandwidth.

- **Mesh:** In a *mesh-based* overlay, such as BitTorrent [36] and PRIME [82], a random directed or undirected overlay among participating peers is formed. Each peer may download some part of the content from any of its neighbors. In contrast to tree-based approach, the mesh-based approach does not need to construct and maintain an explicit overlay structure for delivery of content to all peers. This further simplifies the overlay maintenance in presence of churn.

2.2.1.2.3. Categorizing Content Delivery Overlays Based on Content Delivery Mechanism

According to their content delivery mechanism, content delivery overlays belong to either of the following groups:

- **Push:** In push-based content delivery, often used over tree-based overlays, each parent is responsible for forwarding the content to its children. The content flow to all peers is predetermined with the overlay shape. For instance, SplitStream [23] is a high-bandwidth content distribution system based on application-level multicast. In this application, multiple trees are formed and the shared stream is divided into multiple sub-streams, each pushed down through one of the trees. All proposed end-system multicast projects such as Narada [33], NICE [11] and Overcast [68] also follow the push mechanism.

- **Pull:** In pull-based content-delivery, usually used over mesh-based overlays, peers exchange their block availability status and then each peer requests or *pulls* the blocks it needs from neighbors who have them. With this mechanism, no peer is responsible for providing certain content to another and the data exchanges are based on availability and request. For instance, in BitTorrent[36] peers receive a *bitmap* depicting content availability at each neighbor and then use *rarest-first* policy to decide which blocks to request from their neighbors ensuring maximum block diversity in each neighborhood. Peers will only provide a sustained upload if the receiving party also provides them with a “high” upload rate on the blocks that they request. Non-contributing peers will get *choked* by other peers and may not receive a sustained download.

2.2.2. Characterizing P2P Overlays

Due to the increasing popularity of P2P applications, several research studies are published that try to characterize P2P applications through *(i) network measurement*, *(ii) modeling* and *(iii) simulation*. In this section, we review some outstanding examples of these research studies.

- **Network Measurement:** In this class of studies, Internet measurement is performed over an active P2P overlay in order to assess performance, show possible shortcomings or provide an analytical model.

  Saroiu et al. [116] perform a measurement study on Gnutella and Napster P2P overlays. They measure peer properties including session times and number of shared files, as well as network properties such as end-to-end latency, reported and available bandwidth. Their measurements show that there is significant heterogeneity and lack of cooperation across peers participating in Gnutella and Napster.

Stutzbach et al. [132] introduce *cruiser* a high performance crawler for the Gnutella overlay. Using cruiser they capture full snapshots of the Gnutella overlay taken in a few minutes. They show that snapshots taken with slow crawlers lead to erroneous results biased towards short-lived peers. The authors observe an *Onion-like structure* according to which peer connectivity is related to uptime. Moreover, they show the existence of a *stable core* in Gnutella overlay that ensures reachability despite peer participation dynamics.

Izal et al. [66] provide a measurement study of the BitTorrent using a 5-month long BitTorrent tracker log file. Using this source of information the authors capture several metrics related to a popular swarm including population, each peer's upload and download volumes and rates and downloading times. They show that the seeds (the peers who stay in the system after download completion) significantly contribute to the system and they show that BitTorrent can successfully sustain handle flash crowds.

In our research work presented in Chapter III, we capture a large number of snapshots from the Gnutella overlay during a 15-month time-span. We characterize the evolution of Gnutella during this time period and show how the revisions of the popular Gnutella clients have effectively managed to keep the overlay balanced and efficient despite the population becoming quadrupled.

– **Modeling P2P applications:** Analytical and stochastic modeling is used in a number of research studies, in order to capture and explain some of their characteristics. Qiu and Srikant [98] propose a fluid model of BitTorrent using game theory and validate the model by simulation and experiments. They assign exponential distributions to peer arrival rate, abort rate and departure times and model peer evolution from the joining time until it leaves the system using a fluid model. They provide formulas for the number of seeds and downloaders and downloading time accordingly assuming a Nash equilibrium.

Some other research works also target modeling of different aspects of P2P applications. Ge et al. [50] model a generic P2P file sharing system as a multiple-class closed queuing network. Zou and Ammar [142] provide a "file-centric model" for P2P file sharing systems. In their model, they focus on a file's movement through the system and its interaction with the peers.

None of the observed modeling studies focus on the overlay structure and characteristics.

- **Simulation studies on P2P applications:** Many research studies on P2P applications use some kind of simulation. Simulation is often used as a low cost method for evaluating a proposed system or a modification to an existing system. Simulations may be performed in different levels. A *session-level* simulation of a P2P system provides a simple environment for testing basic functionalities of a P2P system without getting involved in packet-level details and dynamics.

  Bharambe et al. [15] develop a session-level simulator of the BitTorrent system that models peer activity (joins, leaves, block exchanges) as well as many of the associated BitTorrent mechanisms (local rarest first, tit-for-tat). Using their simulator, they study effectiveness of BitTorrent's mechanisms and show that their proposed technique can improve fairness in BitTorrent. As another example, in our study presented in Chapter IV, we perform session-level simulation of an unstructured P2P overlay and our proposed sampling technique, in order to study the effect of churn on the accuracy of sampling.

A *packet-level* simulation is closer to a real experiment. Such simulations are often used in evaluating lower layer protocols such as congestion control, routing and data link layer protocols. However, in the cases that the packet dynamics are important to the applications functionality, they can also be used. The network simulator (NS-2) [65] is widely used by the researchers as a reliable and flexible packet-level simulator. For instance, Magharei et al. [82] implement their proposed P2P streaming application over NS-2 and use it to evaluate its functionality and performance.

Although packet-level simulations are more realistic, they may not be used for simulating very large networks. In this case, session-level simulation may be used if the packet-level details are not very important.

2.3. Internet Topology (Underlay)

Although the Internet is a man made phenomenon, because of its true distributed nature, no entity can claim to have a full map of its topology. Since the rapid evolution of the Internet in the 90s, capturing its topology has become an interesting challenge for the researchers. In addition to the network researchers who study Internet architecture in order to learn the associated features and shortcomings, some scientists have also shown interest in the Internet topology as a large scale complex network.

The Internet topology may be studied in two different levels. In AS-level topology, the connectivity graph is composed of nodes that each represent an

Autonomous System (AS) and edges that represent a physical link between the two corresponding ASs. Roughly speaking, each AS represents an independent company's network and therefore AS-level topology depicts connectivity between companies. Since packet routing over the inter-AS links is handled by the BGP routing protocol and the main deciding factors in BGP routing are often predefined *policies*, having a simple connectivity graph of ASs is of little use when data paths are of any interest. Therefore, the edges of the AS-level connectivity graph are often annotated with the *peering relationships* among the corresponding ASs that also reflect the BGP policies applied on the link.

AS-level topology can provide a high-level view of the Internet and is very useful in describing the structure of the Internet, however, it may not provide enough details about the network technology. In router-level topology, the nodes of the connectivity graph represent routers and each edge of the graph represents a physical link between two routers. The common method to gather router-level connectivity practiced by the researchers is using *traceroute* to capture a massive number of router-level paths.

The main challenges in studying Internet topology are data gathering and characterization. Capturing connectivity data is each level has its own limitations and hurdles which need to be addressed. Once the data is available, a researcher will have to use right methods to look at the data in order to extract new and interesting features.

In this section we survey and categorize the most important research works in Internet topology characterization. In Section 2.3.1. we discuss the studies on AS-level topology and categorize them based on the data sources they have used, characterization method they have employed and also the type of model they provide for the Internet connectivity.

In Section 2.3.2. we survey important research conducted on the router-level topology of the Internet and categorize them according to their data source, characterization method and modeling class.

2.3.1. AS-level Topology of the Internet

The Internet is a network of networks. Each network operated and controlled by a separate and independent administrative entity is called an Autonomous System (AS). Since connectivity structure and packet routing within and among ASs are each based on different goals and principles, AS-level and router-level topology need to be studied separately. Connectivity among ASs is often based on business decisions rather than technical ones and for this reason, packet routing also follows business policies. For instance, a small ISP often chooses a provider offering a lower price for their desired service level.

In the AS-level connectivity graph, each node represents an AS and each edge shows a physical connection between two ASs. Note that if two ASs cover a large geographical area, they may have multiple physical links connecting their networks in different locations. However, in AS-level topology the number of links between two ASs is usually not considered.

One of the main challenges of the studies on the AS-level topology is obtaining a reliable data source. In most of the studies, the AS-level connectivity data is obtained from BGP monitoring and archiving servers such as University of Oregon's RouteViews. Although the data from such sources is known to be incomplete, it is still used as the best source of information on AS-level connectivity. One of the main reasons for studying Internet topology is learning about the paths that the data packets traverse from source to destination. Since the inter-AS routing is policy-based and handled by BGP routing protocol, the BGP policies also need to be included in AS-level topology, otherwise, the connectivity information will not be useful. In Section 2.3.1.1., we discuss different data sources used in AS-level topology and categorize published works from this aspect.

AS peering policies are usually simplified in AS relationships. In this categorization the relationship between each pair of connected ASs belongs to one of the following groups: (i) customer-provider, (ii) peer-peer and (iii) sibling-sibling. The basic BGP policy that is commonly used is usually referred to as "valley-free" routing. This model associates a hierarchical model to the Internet in which each customer is located below its provider(s). In this hierarchy, the top level ASs have no providers, instead they are connected to each other over peer-peer relationships. In this hierarchy, the *tier* of each AS is simply its level in the hierarchy, where top level ASs are tier-1, their customers are tier-2 and so on. This hierarchical structure is an insightful characterization of the AS-level topology. In Section 2.3.1.2., we discuss the characterization techniques and methods used in AS-level topology and categorize the research works from this point of view.

In order to better understand the AS-level topology, some studies have taken the modeling approach. In some of these works, connectivity of the ASs and its pattern and evolution are the subject of mathematical and stochastic models. For example, the node degree distribution of the AS connectivity graph has been modeled with different stochastic models. We will discuss the modeling alternatives and survey research on modeling AS-level topology in Section 2.3.1.3..

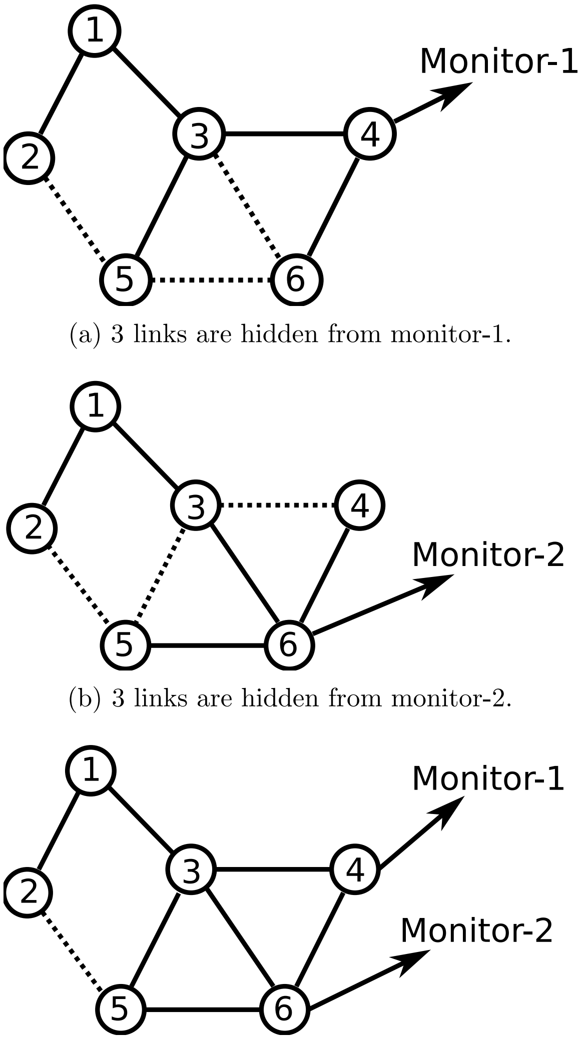
2.3.1.1. Data Sources for AS-level Topology

One of the main challenges of studying AS-level topology of the Internet is obtaining accurate and complete data. Using inaccurate, outdated or biased data can mislead a researcher towards incorrect conclusions. For instance, Chen et al. [28] show that missing a large number of peering links interconnecting medium-sized ISPs in the BGP traces used by some earlier works has been the main cause of observing power law degree distributions and consequent incorrect results.

Research studies have used three sources of data in studying AS-level topology. Below, we discuss these sources and survey the studies using each.

- **Using BGP archives:** One group of common sources of information for capturing AS-level topology are public BGP monitoring and archiving servers. One of the mostly cited such projects is University of Oregon's RouteViews that has been actively monitoring and archiving BGP routing tables and updates since late 1997. In BGP, each routing update includes the complete *AS-path* from the update origin up to the router receiving the update, therefore, each BGP router maintains all the AS-level paths connecting it to all other reachable networks, which is essentially one view of the Internet's AS-level topology. We should note that this view, as shown in Figure 2.1. only includes the links appearing in the paths starting from our BGP router's AS and all other links of the AS connectivity graph are hidden from it.

  In RouteViews, a large number of BGP peerings are established to many volunteer ASs all over the Internet which will act as RouteViews' vantage points. Over these peerings, each vantage point relays all the updates visible from their



(a) 3 links are hidden from monitor-1.

(b) 3 links are hidden from monitor-2.

(c) One link is hidden from both monitors.

FIGURE 2.1. BGP monitors cannot observe all links.

points of view to RouteViews. Effectively, the set of all paths received by RouteViews includes a large portion of all the links between ASs.

Using BGP archives one can produce an AS-level topology snapshot that includes all the active ASs. Also, using saved archives from different points in time, one can study the dynamics of the AS-level topology over a certain time period. On the down side, BGP snapshots often do not include backup links since they are not actively used and advertised by the corresponding ASs unless their main links stop working. Also, as mentioned earlier, by including more vantage points (BGP peers), a BGP monitoring service can extend their sight to observe a larger number of links, however, there are always links that remain hidden. Roughan et al. [112] try to identify and enumerate these missing links.

Several studies on AS-level topology such as papers by Govindan et al. [52], Faloutsos et al. [45], Medina et al. [89], Gao [48] and Mahadevan et al. [85], the authors have used BGP data to produce AS-level topology of the Internet. Chen et al. [28] expose incompleteness of BGP data as the main cause of the observed power-laws in earlier works such as the paper by Faloutsos et al. [45]. They claim that in BGP snapshots 20%-50% of the links are missing.

Chang et al. [24] compare RouteViews data sets with the BGP data sets they have gathered from a set of looking glasses and routing registries and find 25-50% more AS relationships and 2% more ASs. A looking glass is a web interface allowing public viewing access to an ISP's BGP routers, while in an Internet Routing Registry (IRR), the routing policies of each AS is maintained in a public database.

Zhang and Liu [139] compare the AS-level topology obtained from RouteViews snapshots with those they have produced by gathering data from looking glasses, routing registries and multiple route servers including RouteViews. In order to observe backup links that do not usually appear in the RouteViews paths, they obtain all BGP updates over a one year period from RouteViews and include the links observed in these updates to their data set as well. The final AS-level topology they produce includes 44% mode links and 3% more ASs than the average graph obtained from RouteViews data alone.

Roughan et al. [112] aim to estimate the number of missing links in the AS connectivity graph obtained from RouteViews using stochastic and information theory models. Their estimates approve 3 earlier works ([139, 91, 61]) that tried to produce a complete AS-level topology. They estimate the number of missing links to be about 37% of the observed links at a certain time. They also estimate that using 700 route monitors, we can observe 99.9% of the links in the AS connectivity graph.

– **Converting router-level paths to AS-level:** Some researchers including Chang et al. [25] have suggested gathering router-level path information such as traceroute logs and converting them to AS-level paths. In this method, AS-level connectivity can be obtained in finer granularity (*e.g.*, multiple links between ASs). Another advantage in comparison with BGP data is capturing ASs whose routes are aggregated in BGP with other ASs. However, using this method involves some serious data gathering challenges, *i.e.*, accessing a sufficient number of vantage points to run traceroute experiments. Also, traceroute data is known to have certain issues resulting in incomplete or in some cases erroneous data that we will discuss in Section 2.3.2.. Besides these issues, mapping routers to ASs may also add some error due to using foreign IP addresses in border routers. Chang et al. [25] use traceroute logs from the Internet Mapping Project [29]. They present some techniques to avoid the effect of the issues mentioned above. The authors claim that the method addresses some shortcomings of the BGP-based method, however, due to the increasing security concerns, networks are blocking traceroute access to their networks and therefore capturing a nearly complete picture of the Internet using traceroute based methods is infeasible.

– **Extracting AS relationships from routing registries:** One of the services usually provided by the Regional Internet Registries (RIRs), including ARIN[8], RIPE[110], APNIC[7], LACNIC[73] and AfriNIC[1], is maintaining routing registries for their own geographical zones. Additionally, some third party organizations such as RADB[100] also run routing registries. A routing registry is a public database for keeping and publishing the BGP routing policies used by individual ASs over each of their peering relationships with their neighboring ASs.

  Using routing registries, a researcher not only can obtain AS connectivity information, he can also infer the relationship types using the policies listed. Routing registry data can be quite useful for an Internet topology researcher, since it provides all peering information including backup links as well as details of the policy without any measurement and these information are hard to obtain from sources discussed earlier. However, in practice routing registry data is of limited use in Internet topology studies because the entries are often out of date and incomplete due to the fact that the ISPs have little motivation to keep them up to date. Gao [48] uses ARIN's routing registry information to compare and evaluate her algorithm for inferring AS relationships while Chang et al. [24] use RIPE's routing registry to complement data obtained from RouteViews and looking glass websites, as described earlier in this section.

  In summary, although routing registry data is often considered incomplete and therefore it is not relied on as a sole source of information, some researchers have used it in order to complete the topology obtained from other sources or as a reference to compare and evaluate their methods, such as inferred sibling relationships.

2.3.1.2. Characterization Methods for AS-level Topology

Characterizing the AS-level topology of the Internet is a common goal that has been pursued using different techniques and methods. The common goal is discovering interesting characteristics and features of the AS connectivity graph that can provide an insight on better understanding the way Internet works and evolves. The most important subjects of AS-level topology characterization, according to the volume of research work, are : (i) Degree distribution in AS connectivity graph, (ii) Hierarchy of the AS-level topology, and (iii) Inferring inter-AS relationships. In this section we survey the research work addressing each of these subjects and mention the benefits and the challenges involved in each case.

– **Node degrees in AS connectivity graph:** In AS connectivity graph, each node, representing an AS, is connected to a number of other nodes. The number of ASs that each AS is connected to, determines the degree of the corresponding node. Node degree distribution provides the most basic view of a graph's connectivity and therefore it has been used in numerous research works to capture and present the structure of the AS connectivity graph.

  Faloutsos et al. [45] in one of the first works in AS-level topology characterization claim that the degree distribution of the AS connectivity graph follows a power-law distribution. They also discover other power law relationships in the Internet topology, including the number of nodes within h hops as a function of h. Based on these finding, they show that random graphs do not represent the Internet topology. Later, Medina et al. [89] analyze possible root causes for the observed power laws in the previous work. They identify *preferential connectivity* together with *incremental growth* as the key contributing factors to the power law relationships. Fabrikant et al. [44] propose an explanation for the power laws based on a toy model of Internet growth in which two objectives are optimized simultaneously: last mile connection costs and transmission delays measured in hops. Power laws tend to arise as a result of complex, multi-objective optimization.

Chen et al. [28] identify incompleteness of BGP data as the main cause of the observed power-laws. The paper shows that by compensating for the missing links, the resulting degree distribution becomes heavy-tailed but not power-law. It also claims that the connectivity dynamics and growth processes assumed in [89] do not apply to the Internet. Later, Li et al. [76] show that degree distribution alone cannot capture the specifications of a graph completely by showing examples of different graphs with very different characteristics showing the same degree distributions. Although the heavy-tail degree distribution of the AS-level topology shows that there are a few ASs with very large degrees while the vast majority have very small degrees, such pattern should not be used to conclude a certain structure in the Internet.

Joint degree distribution is proposed by Mahadevan et al. [85] as a definitive metric in order to capture the connectivity preferences with regards to node degrees. This paper shows that the Internet topology is disassortative, *i.e.*, nodes have a tendency for connecting to nodes with dissimilar degrees.

Caveats: Although node degree is an important factor and it can reveal several features of the graph, care must be taken in identifying graphs based on their degree distributions alone. Also, considering the incompleteness of the available AS connectivity graphs, any findings regarding node degree may be an artifact of the missing links.

- **Inferring relationships between ASs:** As mentioned earlier, AS relationship information is an important part of the AS-level topology since such information is necessary in order to understand the BGP policies that are used in routing among ASs. However, the relationships are business information and can be private. Therefore, the researchers have tried to infer the relationships from other information such as AS connectivity graph as well as the routing registry information. These inference techniques are often based on common conditions that one expects to observe in these relationships. For instance, it is expected that the degree of a provider be larger than that of its customer. However, since there are always exceptions and special cases, such assumptions lead to a certain amount of error. Although inferring customer-provider relationships might be less challenging, inferring peer-peer and sibling-sibling relationships often requires additional information.

  Gao [48] proposes a method for inferring relationships from the AS-paths obtained from RouteViews. Her proposed algorithm is based on the *valley-free* routing principle according to which no customer lies between two providers of its own in an AS-path since a customer does not provide transit service to its providers. It is also assumed that in each customer-provider relationship, the degree of the provider is larger than that of the customer. In this algorithm, in each AS-path the AS with the highest degree is chosen as the top AS and the other relationships are inferred based on the valley-free principle. By processing each path, one vote is cast towards the inferred relationships along that path and the final decision is based on the total votes resulting from processing all the paths after certain adjustment and refinement. Subramanian et al. [133] present the AS relationship assignment as an optimization problem and propose a heuristic algorithm to solve this problem by combining AS-paths from multiple vantage points in the Internet. Other works by Xia and Gao [136] and Dimitropoulos et al. [41] evaluate the proposed algorithms and suggest incremental improvements over those algorithms by accounting for missing relationships and including routing registry information in inferring sibling relationships, respectively. The Cooperative Association for Internet Data Analysis (CAIDA)[20] generates AS relationship snapshots of the Internet using algorithms from [41] applied to RouteViews data on a regular basis and archives and publishes the results for the public use.

The main challenges involved in this problem are inferring sibling-sibling relationships as well as accounting for the missing connectivity information. The inferred relationships are widely used in a variety of research works involving the Internet topology and traffic.

- **Hierarchy of the AS-level topology** The relationships between ASs are commonly used in the area of AS-level topology to depict the hierarchy of the Internet. According to this hierarchy, each AS is assigned a tier number reflecting a level of the hierarchy. Generally, tier-1 ASs are those who have no providers and a tier-n AS has at least one tier-$(n-1)$ provider. Understanding the hierarchical structure of the Internet in insightful and can be used to explain many characteristics of the Internet and the way traffic flows over it. However, there are a few challenges that makes this work nontrivial. First, there is some controversy on defining tier-1 ASs and the instances. Since the business contracts among top-level ASs are confidential, accurate inference of the type of relationships becomes challenging. Also, some peer-peer and sibling-sibling relationships make shortcuts linking different tiers to each other that makes some of the assumptions invalid.

Ge et al. [49] provide an algorithm to classify ASs in their respective tiers according to the inferred customer-provider relationships using the above definition. They also make available a tool called *TierClassify*) implementing their algorithm for public use.

Dimitropoulos et al. [40] provides an alternative classification of the ASs. They define 6 classes of ASs, namely, Large ISPs, Small ISPs, Customer ASs, Universities, Internet exchange points and Network information centers. They use AdaBoost machine learning tool and manually classify more than 1000 ASs in order for the machine learning algorithm to learn the characteristics of each class. The AS attributes include IP space size and type and number of AS relationships along with boolean attributes that reflect the results of searching certain words in the AS description field from the registry.

2.3.1.3. Modeling the AS-level Topology

Modeling is often used in characterizing complex systems. This method can be very helpful in simplifying and understanding the basic rules governing the system behavior and it can possibly enable the researchers to predict the system's behavior in response to anomalies or unexpected events. Modeling the AS-level topology has been pursued in a number of research works. The main challenge of a useful modeling work would be finding the right model that not only fits measured data but also can provide an insight into the limitations and tradeoffs governing the AS-level topology.

In this section we categorize some of the research works on modeling the AS-level topology according to the type of the model they use.

- **Descriptive Models:** In this class of modeling works, certain characteristics of the AS-level topology are captured by measurement and then mathematical models are provided trying to fit the captured data. Faloutsos et al. [45] fit the degree distribution of the AS connectivity graph with a power-law model and Medina et al. [89] find *preferential connectivity* and *incremental growth* as the main causes of the power-laws. However, Chen et al. [28] questions the Barabasi-Albert model for AS-level topology based on the fact that the observed degree distributions were artifacts of incompleteness of the AS connectivity graph. They suggest that the actual degree distribution of the AS connectivity graph does not fit a BA model although it is heavy-tailed and suggest adapting a HOT-based model for AS-level topology.

  In another example of modeling, Roughan et al. [112] in an effort to discover the missing links of the AS-level topology, employs the capture-recapture idea from biology, to derive a Binomial Mixture Model(BMM) for the number of observations of each link across all view points. They estimate the model parameters using an Expectation Maximization (EM) algorithm.

  Since descriptive models are only based on measured data, they can be vulnerable to measurement errors.

- **Generative Models:** In multiple areas of networking, the researchers need to set up simulations. These simulation often need a topology graph that has similar characteristics as the Internet. Generative models are algorithms designed to generate graphs with similar characteristics as the modeled graph. BRITE [88]is a topology generator tool that is able to generate topology graphs using a variety of models. In particular *ASWaxman* generates AS-level topologies with the properties of a random graph in which nodes with shorter distance are more likely to get connected to each other while *ASBarabasiAlbert* results in topologies that have power-law degree distribution and try to represent the hierarchical structure of the Internet. Some older examples of the generative models of the Internet are *GT-ITM and Transit-stud*[22] and *Tiers*[42]. A representative generative model can be a quite useful tool for evaluating a design using simulation or verifying a hypothesis about the Internet however since each model focuses on representing the Internet from a certain aspect or a number of aspects they always miss some other characteristics of the real Internet.

– **HOT-based Models:** Any complex system can be thought of as a solution to an optimization problem with certain constraints and tradeoffs. Highly Optimized Tolerance (HOT) denotes a class of models that are based this very principle. In HOT-based modeling, researchers try to find use these optimizations and tradeoffs in order to build a model that describes behavior of the system. Fabrikant et al. [44] are the first to provide a HOT-based model of the AS-level topology. They propose a toy model of the incremental access network design optimizing a tradeoff between connectivity distance and node centrality. They also show that the relative importance of these factors can significantly change the resulting topology. Alderson et al. [4] make a proposal of identifying the economic and technical tradeoffs involved in network access design for building a HOT-based model of the Internet topology. They suggest that the "Buy-at-Bulk" scheme is an optimization to a tradeoff on the bandwidth provisioning problem according to which "larger capacity cables have higher overhead costs , but lower per-bandwidth usage costs."

  Chang et al. [26] also apply HOT concept to the AS connectivity problem. They extend earlier works by presenting a multivariate optimization problem that determines AS decisions in choosing an upstream provider: (i) AS-geography *i.e.*, location and number of ASs within each AS, (ii) AS-specific business models and (iii) AS evolution *i.e.*, a historic account of each AS in the dynamic market. Although HOT-based models are much more challenging to develop compared to the descriptive models, they are quite more robust against measurement errors. On the other hand, since the Internet evolution is a distributed process driven by many independent entities with potentially different goals and limitations, assuming that the same set of tradeoffs are controlling this process in different places seems questionable.

2.3.2. Router-level Topology of the Internet

As mentioned earlier, the Internet topology studied in two different abstraction levels. While in AS-level topology the connectivity among ISPs is the focal point and the most important factor forming the topology is business relationships, in router-level topology, the network infrastructure is the primary subject of study and the network technology is the major factor.

The most important challenge in studying router-level topology of the Internet is data gathering. Although a simple tool such as *traceroute* is potentially able to capture the router-level paths between any two points in the Internet, practical limitations significantly reduce the usability of the results. Commonly in router-level Internet topology, the main source of information is the data resulting from of the Internet is captured via a large-scale series of *traceroute* operations.

The router-level topology, if captured with acceptable accuracy, provides a higher resolution over AS-level topology. In spite of the AS-level topology, multiple paths may be captured as well. However, the main problem remains accurate data gathering due to limitations that the traceroute and other tools have.

2.3.2.1. Data Sources for Router-level Topology

In this class of studies, *traceroute* has been the basic tool used when a global scope is desired while in studies with local scope, topology information is usually provided by the ISPs. Traceroute[67] can provide the router-level path from a source over which the researcher has control to any arbitrary destination host over the Internet. Traceroute, originally developed by Van Jacobson, sends a series of packets with controlled TTL values. TTL (time to live) of an IP packet determines the maximum number of routers it can pass before reaching destination, a mechanism designed to dispose of the packets that get stuck in routing loops as a result of routing problems. In order to capture the router-level path from host A to B, traceroute must be run on host A. Upon execution, it starts sending packets (ICMP or UDP packets depending on version and parameters used) with TTL value of zero. As a result, the first router on the path will dispose of the packet and send an ICMP error message back to the sender. In each round, traceroute increments the TTL value by one until either the packet reaches the destination or the TTL reaches a predefined maximum value (usually 30). The error messages returned by the transit routers as a result of TTL expiration are used by traceroute to identify routers on the path and thereby produce a list of IP addresses of the routers on the path.

Although traceroute has been very useful for determining routing problems, its ability to capture the global router-level topology is limited for the following reasons. First, in order to produce the Internet topology a researcher needs to capture a large number of paths. The usefulness and representativeness of the resulting topology highly depends on the number and distribution of the endpoints of the paths. In order to capture the Internet topology with an acceptable coverage, a researcher would need access to a large number of hosts worldwide which is very hard to obtain. Second, increasingly many networks are using firewalls that block traceroute packets into their networks, specially at the edge of the Internet. This will limit the coverage and accuracy of the captured paths and the resulting topology. Third, there are known limitations in traceroute technique that result in erroneous results in presence of dynamic routing. Remember that each hop is identified by a separate packet and due to dynamics of routing, different packets may take different paths between the same pair of end-points. Using such erroneous paths can mislead the researcher to including false links in the topology.

There are a number of projects and tools built on top of the basic traceroute technology with the goal of achieving higher accuracy and wider coverage. Since data gathering is a major challenge in studying the Internet topology, below we compare these data gathering tools and projects and the research works using each.

– **Skitter**[21] is a project of CAIDA with the goals of (i) determining forward IP paths, (ii) measuring RTTs, (iii) tracking persistent routing changes and (iv) visualizing network connectivity.Skitter uses the traceroute technique in addition to some kernel hacks in order to increase the accuracy of RTT measurements. Barford et al. [14]employ Skitter traces between 8 sources and more than 1000 destinations spread all over the world to build up a partial picture of the Internet backbone in the year 2000. While the sources are all hosts owned by the project placed in volunteer networks, the destinations are a web servers distributed over the Internet. They argue that towards the goal of characterizing the Internet backbone, the utility of adding more vantage points in a traceroute study is marginal. Specifically, they claim that a careful selection of two or three vantage points will result in nearly same coverage as all the 8 sources used by skitter. *Archipelago (Ark)* is the evolution of the skitter project including the skitter monitors, measurement tool, several other data processing tools. Later the skitter measurement tool was replaced with *scamper*. Scamper [54]is an extended version of the skitter tool that also supports IPV6 and is able to flexibly use TCP or UDP probing packets. Luckie et al. [79] use scamper from 8 vantage points distributed across the globe and 3 different sets of destinations including random routable addresses, top 500 websites according to Alexa [6], and a list of known routers from an earlier study. They show that although ICMP traceroute probing is able to reach more destinations and discover more AS links, UDP probes infer the greatest number of IP links.

– **Mercator** proposed by Govindan et al. [53] focuses on the problem of finding useful destination addresses in a traceroute-based technique. They use *informed random address probing* to make guesses about which prefixes might contain addressable nodes by heuristics from common patterns of IP space allocation. They also employ source routing (supported by a only 8% of the Internet routers) to include cross links considering that they only employ one vantage point. Mercator addresses the problem of IP address aliasing by sending probes to the discovered address of the router and comparing the discovered address with the responding address to verify whether or not the two addresses belong to the same router.

– **Rocketfuel** proposed by Spring et al. [125], is a tool for mapping the router-level topology of an ISP using traceroute, RouteViews data, and reverse DNS. They perform traceroute experiments sourced from 800 vantage points hosted by nearly 300 traceroute web servers (servers that provide traceroute service from their location to any desirable host). The authors focus on improving the efficiency of probing. Using their path reduction techniques, they manage to reduce the number of probes needed by three orders of magnitude compared to a brute-force all-to-all probing without any significant accuracy loss. They capture a much more complete graph with roughly seven times as many links.

- **Paris Traceroute**was proposed by Augustin et al. [9]. The authors focus on the traceroute errors in presence of dynamic routing. They list possible traceroute anomalies such as loops, cycles and diamonds and show how they can happen as a result of different forms of dynamic routing such as load balancing.

Also, in a number of studies, such as the work by Li et al. [76], Abilene is used as a source of data. Abilene Network (now known as Internet2 Network) is a high performance backbone network in the U.S. mainly connecting academic and research centers throughout the country using high speed links (10 Gbps). Abilene Network provides a useful research case for Internet researchers because it makes all the topology and traffic information publicly available. Although such data does not represent the Internet, it still provides useful insights particularly as a real testbed to evaluate methods and tools for measurement and characterization of the Internet topology and traffic.

Despite all the efforts, finding a reliable source of date that provides highly accurate and representative data on the router-level topology of Internet is still a problem.

2.3.2.2. Characterization Methods for Router-level Topology

In characterizing router-level topology, the researchers usually search for interesting characteristics and features of the routers' connectivity. In some research works with pure science theme, the router connectivity graph is studied as a complex network with little attention to the context and the root causes. Another group of works study the router-level connectivity in order to understand the Internet and possibly discover its features and shortcomings.

Below, we survey some of the subjects discussed in the router-level topology characterization and discuss their advantages and shortcomings.

– **Node degree distribution** is commonly used as a characterization metric for router-level topology of the Internet. Similar to many other graphs, degree distribution is often considered the most basic piece of information that can capture and present some characteristics of the router-level connectivity graph mainly by showing the heterogeneity level across the nodes. Faloutsos et al. in their SIGCOMM paper [45] in addition to the AS-level topology that we discussed in Section 2.3.1., use a router-level topology map from an earlier work from 1995 and show that the degree distribution follows power law similar to the AS-level topology. This result has been rejected from different aspects in the works published later. Yook et al. [138] suggest a fractal model for the Internet topology and show that the power laws do not represent the Internet and the degree distributions are in fact exponential. Lakhina et al. [74] show that the power laws are an artifact of sampling the router-level topology using traceroute. They perform simulations showing that traceroute-like sampling will result in power-law degree distributions even if the original graph is a random ER graph.

Nonetheless, all the studies agree that the router-level topology of the Internet has a heavy-tailed distribution. Some papers such as the work by Albert et al. in the Nature journal [3] have warned that in this heavy-tailed distribution, there are extremely high degree nodes that act as the central hubs of the Internet and failure of each can disconnect a large portion of the network. This idea was rejected by Li et al. [76] who showed that several graphs with very different characteristics may have similar power-law degree distributions. Although degree distribution provides a first level of understanding about the router connectivity graph, care must be taken not to read too much from it.

– **Tomography of the Internet** Since the researchers do not have direct access to the core of the Internet, they use information gathered from several endpoints in order to provide an image of the core. This practice is commonly referred to as *tomography* in many disciplines. According to this definition, we can categorize any traceroute-based studies of the Internet router-level topology as tomography. Coates et al. [35] provide a survey of the techniques for making inferences about the Internet based on the observed behavior. They include two classes of network tomography: (i) estimating link-level characteristics from path-level data and (ii) estimating path-level characteristics from link-level data. The inferred data may be loss rate, packet delay or the connectivity. Although the common perception is that having more vantage points, the tomography results will be more accurate and reliable, Barford et al. [14] question the "more is better" approach and show that increasing the size of the network measurement infrastructure only leads to marginal improvement in Internet tomography.

2.3.2.3. Modeling the Router-level Topology

The modeling approaches for router-level topology of the Internet are similar to those we discussed for the AS-level topology in Section 2.3.1.3.. They aim to find mathematical models that describe and explain the connectivity patterns. A good model not only should match the measured and confirmed data from the router-level topology, it should also provide an insight for understanding how the network grows and evolves. Using a reliable model, a researcher can detect vulnerabilities or predict potential malfunctioning threatening the Internet. Developing models that bear the mentioned capabilities has been a challenging task. Below we provide a survey of the modeling studies on the router-level topology of the Internet.

- **Descriptive Models:** In this group of studies, certain measured data on the router-level topology is examined for similarities against known mathematical models. Faloutsos et al. [45] use a router-level topology data set captured in 1995 and find similarities with the power-law model which was later rejected. This work is explained in more detail in Section 2.3.1.3..

- **Generative Models:** Similar to the description given for AS-level topology modeling, generative models of the router-level topology are algorithms or programs designed to generate synthetic topologies resembling the real router-level topology of the Internet. These models are widely used in simulation-based evaluation of network applications. Furthermore, they often provide the flexibility of generating a range of topologies by one or more controlling parameters.

  BRITE [88] (also an AS-level topology generators) is a tool that is able to generate topology graphs using a variety of models. In the class of *flat router-*

level models, it places the nodes on a plane based on random or heavy-tailed model and after establishing the links using either *RouterWaxman* or *RouterBarabasiAlbert*, assigns link bandwidth according to either constant, uniform, exponential or heavy-tailed models with controlled parameters. *RouterWaxman* generates AS-level topologies with the properties of a random graph in which nodes with shorter distance are more likely to get connected to each other while *RouterBarabasiAlbert* results in topologies with power-law degree distribution by using *incremental growth* technique with *preferential attachment*. Medina et al. [88] also categorize earlier generative models into two groups of (i) *ad-hoc models* that are mostly built based on educated guesses such as the hierarchical structure of the Internet (*e.g.*, GT-ITM [22]) and (ii) *measurement-based models* that try to reproduce the measurement results such as Barabasi-Albert models that reproduce power-law degree distributions using preferential attachment.

Li et al. [76]use a first-principles approach in developing a generative model for the router-level topology of the Internet. They apply technological limitations and economical considerations into a performance optimizing design process yielding a generative model of the Internet's router-level topology.

While a generative model can be useful in evaluating a new design using simulation or verifying a hypothesis about the Internet structure, one may not assume that a generated topology resembles the network in every aspect. Limitations of each generative model should be recognized before employing them.

– **HOT-based Models:** General description of HOT-based models is provided in Section 2.3.1.3.. Li et al. [76]pursue a first-principles approach aligned with the idea of HOT-based modeling in which the technological constraints and economical considerations are identified as the primary factors determining a network's decisions at the time of topology construction. According to this paper, the router building technology limits the bandwidth-degree product due to the limited bandwidth of the router's data bus. They use a number of state of the art Cisco routers in 2004 in order to identify the technological limits at the time and argue that the market mostly demands for relatively low bandwidth ports while the core of the network requires very high bandwidth ports. Therefore the solution to the optimization problem would be configuring routers with maximum number of ports at the edge (low bandwidth) and maximum bandwidth ports (small number) at the core of the network. They compare graphs generated by different generative models and show that the HOT graph has the highest performance (throughput) and lowest likelihood. The authors publish another paper [5] in which they extend the previous work by evaluating their HOT graph with Abilene and Rocketfuel data. HOT-based models are still a hot topic in studying the Internet topology and due to not relying on measurements, they are not subject to measurement errors. However, it seems that the idea is not yet developed enough to produce useful models representing the Internet topology from multiple aspects.

Yook et al. [138] suggest a fractal model (scale-free) for the Internet topology in which the links are placed by competition between preferential attachment and linear distance dependence. According to their scale-free model, the Internet connectivity depends on a small number of very high degree nodes that representing the Internet hubs. They conclude that although the Internet is robust to random node failures, it is quite fragile to targeted attacks on these hubs. Doyle et al. [43] extend their earlier works on modeling the router-level topology by suggesting a "robust-yet-fragile" model for the Internet. They show that the characteristics of the scale-free model does not match those of the Internet while the HOT model they had suggested earlier [5] shows similar features and characteristics as the Internet using the two metrics of *performance* and *likelihood*. In their view, the Internet's fragility does not lie directly within its topological aspect. By focusing on the protocol stack, they mention that the lowest layers of the Internet are highly constrained by technological and economical limitations while the higher layers have more flexibility and freedom. The flexibility on the higher levels of the protocol stack such as the application layer is what makes the Internet robust and yet the same flexibility makes the network fragile to malicious exploitation.

2.4. Interactions between Overlay and Underlay

In this section we focus on the mutual effects, interactions and possible cooperation between the P2P overlay and the Internet underlay. A number of research studies have focused on the impact of the P2P overlays on the underlying network using measurement and simulation. We discuss this group of studies in Section 2.4.1.. In Section 2.4.2., we discuss the unilateral efforts by the ISPs in limiting the impact of the P2P overlays and the network neutrality concept. Next, in Section 2.4.3., we overview a number of research studies proposing P2P overlays that try to minimize their impact on the underlying network, called ISP-friendly or network-aware overlays. Finally, Section 2.4.4., introduces a number of research projects and engineering efforts proposing cooperation between the overlay and the underlay in order to build overlays that are desirable for both underlay and the P2P application.

2.4.1. Overlay Impact on the Underlay

The direct effect of the overlay network on the underlay is the *traffic* associated with the P2P overlay that can lay a costly and unexpected load on the ISPs. As we discussed in detail in Section II, the P2P traffic in costly for the ISPs because of its temporal pattern and symmetric load. In this section, we survey two research studies that try to characterize the impact of the P2P overlay on the ISPs. They both rely on packet traces captured at vantage points connecting an ISP or campus network to the Internet. They both show that the P2P traffic consumes a large portion of the gateway links and thereby they motivate modifications in the P2P overlays by localizing or caching in order to save a considerable amount of traffic on the Internet gateways of the ISPs.

- Karagiannis et al. [71] compare the load on the ISP for the cases of traditional client-server, P2P, local caching and their proposed mechanism. They propose a locality-aware overlay for peer-assisted content-delivery and show that its performance and the external load (impact on the ISP) is the best among the compared cases. They show that current P2P content distribution overlays (*e.g.*, BitTorrent) are not ISP-friendly because they generate a large amount of external traffic that can be avoided.

- Gummadi et al. [55] capture a 200-day trace of KazaA traffic at their campus gateway. They observe that most requests are for small files while most of the traffic volume is formed by large files. Although they do not capture the internal traffic, they show that there is a considerable amount of requests going outside the network while they can be resolved locally and therefore they suggest that a locality-aware scheme can help in reducing external traffic of KazaA.

2.4.2. Underlay Limiting Overlay

The ISPs have tried to control the P2P traffic in different ways. Toward this end, the P2P overlay traffic needs to be identified first. The simple methods of using TCP and UDP port numbers is now of little use because most P2P applications are flexible in the port number that they use and in some cases NAT traversal techniques - which are now very common - require using non-standard ports. There are a number of commercial protocol analyzers that combine a variety of techniques in order to identify the application responsible for each flow of traffic. These technique include deep packet inspection and traffic pattern analysis. Some researchers including Suh et al. [134] and Branch et al. [19] propose techniques based on temporal patterns of packets and packet sizes to identify Skype traffic.

The next step after identifying a P2P flow would be applying some type of restriction. The following methods have been reportedly used to contain or block the P2P traffic:

– *Packet Filtering:* This method requires implementation on routers, can be costly and limiting the router performance. In this method all packets identified as the target class will be dropped by the router. It will quickly alarm the users because their P2P applications will often stop working and therefore it is rarely used by ISPs who have to compete customer satisfaction. This method has been used in some campus residential networks where the users have limited options.

– *Traffic Policing:* This method (also known as rate limiting) also requires implementation on the routers however it is more flexible and less likely to be noticed by user. In this method, the network administrator defines an access-list that identifies target traffic flows and then assigns a maximum data rate to each class of traffic or to any flow belonging to that class. All packets exceeding the predefined maximum rates, will be dropped and as a result users will experience slow P2P transactions. Since the low speed may be associated with many factors including the P2P application itself, this method does not alert most of the P2P users against the ISP. Class based rate limiting can be technologically costly for the ISPs.

– *Connection Resetting:* This method has been reportedly used by some ISPs and its advantage is that the intervening device does not need to be on the path of the traffic therefore it can be implemented on a regular computer (not necessarily a router) with monitoring access to the traffic. In this technique after detecting a P2P flow, in order to terminate the connection, TCP reset packets are sent to both ends of the connection on behalf of the other end. In order to avoid alarming the users, this method can be applied on a random subset of the matching flows.

– *Transparent traffic redirection:* In this method, designed for localizing BitTorrent-like traffic, the ISP runs a transparent tracker proxy. When BitTorrent clients try to access a tracker to join a swarm, the connection will be redirected to the transparent proxy. The proxy server then controls the external traffic related to the swarm by connecting the local peers to each other and preventing local peers to connect to external peers. This method aims at smoothly limiting of the external traffic with minimum service degradation for the P2P application. However, in BitTorrent-like P2P overlays, certain level of random connectivity is needed to ensure that the blocks can diffuse all neighborhoods. Excessive localization will result in a heavily clustered overlay and thus may degrade the performance of the overlay by limiting the opportunity for peers to help each other.

2.4.2.1. Network Neutrality

All the methods described above, regardless of the technique used, are criticized by a large group of people in the networking community. They believe that the network should treat all packets equally regardless of the application they belong to. In other words the network should avoid discrimination among applications. This thesis, consistent with the end-to-end argument, is referred to as *network neutrality* and was the basis of the FCC's ruling against Comcast[37]. In this ruling, the Federal Communications Commission ordered Comcast, a large ISP with a national market in the U.S., to "end discriminatory network management practices".

2.4.3. Topology Aware Overlays

In response to the ISP concerns, the P2P research community proposed ideas towards ISP-friendly P2P applications. The common goal across these research works is trying to decrease the inter-ISP traffic by increasing the relative number of local P2P connections and reducing number of external connections. Although these methods are often successful in limiting the ISP load, the effect on P2P performance is not evaluated from a neutral point of view. Additionally, since there is no authoritative topology or link cost information, such systems cannot use low cost external links or unpaid peering links between ISPs. In this section, we survey some outstanding research works on this topic.

- Ratnasamy et al. [109] suggest a binning scheme to find nearby nodes for peering and server selection. The bins are formed by sorted closeness to well-known landmarks(*e.g.*, 12 root DNS servers). They assign coordinates to each node in n-dimensional space where each dimension can take 3 values. The authors suggest a modification on CAN to selection node coordinates based on its network location.

- Harvey et al. [59] present *SkipNet*, a DHT-like overlay that allows for content locality and path locality. The locality is based on the node's DNS domain name.

- Kim and Chon [72] present a topologically-aware application-layer multicast overlay. In their scheme, close-by nodes are using network distance measurements to a few landmarks. Nodes are partitioned into topologically-aware clusters and local paths are determined between local nodes.

- Choffnes and Bustamante [31] propose *Ono*, . In Ono, nearby peers are identified according to their CDN server choice. In the CDNs they use, including Akamai and Limelight, a smart DNS server designates closest CDN server to each peer by a DNS lookup. Ono takes advantage of this system and tries to connect peers with the same CDN server together in order to *(i)* reduce the load on external ISP links, and *(ii)* improve system performance by avoiding bandwidth bottlenecks in the network. The authors claim average improvements of between 30% to 200% in download rates on the BitTorrent clients using Ono plugin. An advantage to previous works is that Ono does not need any costly network measurement or probing, instead, it only depends on periodic DNS lookups.

2.4.4. Cooperation between Overlay and Underlay

Considering the limitations of independent (unilateral) ISP-friendly P2P applications described earlier, it has become evident that the proper way to make the applications ISP-friendly is by using information provided by the ISP. *P4P* and *Oracle* were recently proposed based on the idea of an interface between the ISP and the P2P application over which the ISP shares information with the application regarding the ISP's relative preference among candidate peers. In addition to the mentioned research works, there have been ongoing efforts in IETF on the idea of Application Layer Traffic Optimization (ALTO). As a result, multiple Internet drafts have been published addressing different aspects of the problem and their proposed solutions. Below we provide an overview of the outstanding publications on this topic.

– Xie et al. [137] propose *P4P*, an interface that provides ISP preferences to the application layer in order to enable the application to redirect its traffic to satisfy the ISP preferences in its neighbor selection. In P4P, the ISP runs a server called iTracker which is aware of the ISP's topology, current link loads and costs associated to each link. The iTracker is then responsible for translating these factors into a single *cost* metric that can be looked up on a per-destination basis. The local application tracker (*e.g.*, BitTorrent tracker) should contact the iTracker to look up the cost values and include the ISP goals as well as the application goals in the neighbor selection process. The paper also demonstrates, using simulation and experiments that the method improves or at least maintains application performance while reducing the cost on the ISP.

– Aggarwal et al. [2] propose *Oracle*, an interface between ISP and P2P application that takes the list of prospective neighbors from each peer, sorts them according to the ISP preferences before they are returned to the peer. This method is simpler however it requires implementation in each application. Also, the scalability is questionable since the neighbor selection duty is on the shoulder of one server for each ISP.

– ALTO is a working group in the Internet Engineering Task Force (IETF) with the goal of "designing and specifying an Application-Layer Traffic Optimization (ALTO) service that will provide applications with information to perform better-than-random initial peer selection". Here we provide an overview of three Internet drafts published within this working group.

Seedorf and Burger [117] provide a problem statement of the application layer traffic optimization problem. According to their draft, in current P2P applications, peers choose neighbors without reliable information (*e.g.*, based on measurements or simply randomly) leading to suboptimal choices. This document describes problems related to optimizing traffic generated by peer-to-peer applications and associated issues. Such optimization problems arise in the use of network-layer information. Crowley [38] argues that the problem of P2P traffic optimization is not solved by standardization at this point due to lack of motivation in the user community. He suggests that ISPs should deploy pricing models based on the amount of each user's external traffic. Shalunov et al. [119] discuss the format and standardization of the ISP-P2P information export service. The suggested method is similar to P4P[137] and an ISP controlled agent sets priority values on each potential peering relationship. The peers will then select their neighbors according to their own preference, as well as the ISP's.

2.5. Summary

In this survey, we reviewed a number of important research studies on the P2P overlays, the underlying network, and their mutual impacts on each other. We cover a set of fundamental design and evaluation issues by surveying previous studies. We find and report an array of open problems and challenges in the covered area.

In Section 2.3., we studied research works on the AS-level and router-level topology on the Internet. We observed that one important challenge in studying Internet topology is gathering data that is reasonably complete. In studying AS-level topology, the hidden links between low-tiered ASs cause incomplete topology snapshots while in studying router-level topology, limitations of traceroute technique and blocking of probe packets cause incompleteness of the data.

In Section 2.2., research works on P2P overlays were surveyed. We categorized P2P overlays according to their function, structure, shape and content type. These differences have key importance when we study the mutual effects between the underlay and the overlay. One main challenge in P2P overlays is providing incentives for the users to contribute their resources. Towards this end, P2P applications should be designed with selfishness as a basis rather than depending on people's altruism. We observe that although several research works have been published on characterizing P2P applications, the attention on the overlay structure, specifically in modeling areas, has not been significant. Modeling of P2P overlays and their traffic is an important prerequisite for understanting the impact of the overlays on the underlay.

Finally, in Section 2.4., we provided a survey of the research and engineering efforts on the issues involving both the P2P overlay, and the underlying network. We observed that although there are methods proposed for network aware overlay construction with the cooperation of network layer, they are not widely deployed by the ISPs and P2P applications due to the lack of motivation on the user's side which depends on the P2P application performance. There is little unbiased study reporting significant benefits of such cooperation for the user and the P2P application. On the other hand, ISPs still have concerns about the possible abuses and vulnerabilities resulting from an ISP-P2P interface such as P4P.

CHAPTER III

MEASUREMENT STUDY ON GNUTELLA OVERLAY

Most of the content of this chapter has been adopted from my previously published paper [102] co-authored with Dr. Daniel Stutzbach and Prof. Reza Rejaie. The experimental work is entirely mine and the text has been contributed by myself and the co-authors. The Gnutella crawler used was originally developed by Dr. Daniel Stutzbach.

Contrary to common assumptions about the limited scalability of unstructured Peer-to-Peer (P2P) file-sharing applications, the top-three P2P file sharing applications (*i.e.*, FastTrack or Kazaa, Gnutella and eDonkey) have witnessed a dramatic increase in their popularity during the past few years. For example, the number of simultaneous users in the Gnutella network has quadrupled during the 15 months measurement period. Furthermore, some studies report that the popular P2P file sharing applications make a significant contribution to total Internet traffic [124, 70].

To scale with this rapid growth in user population, major P2P file sharing applications adopted a two-tier overlay topology along with more efficient search mechanisms (*e.g.*, Dynamic Querying [47] in Gnutella). In this two-tier overlay architecture, a small subset of participating peers promote themselves to become *ultrapeers* in a demand-driven fashion and form a *top-level* overlay. Other peers, called *leaf peers*, connect to the top-level overlay through one or multiple ultrapeers (Figure 3.1.). The two-tier architecture attempts to dynamically maintain the following two properties in order to scale with the number of peers while ensuring short pairwise distances between peers as they join/leave the system: *(i)* a proper

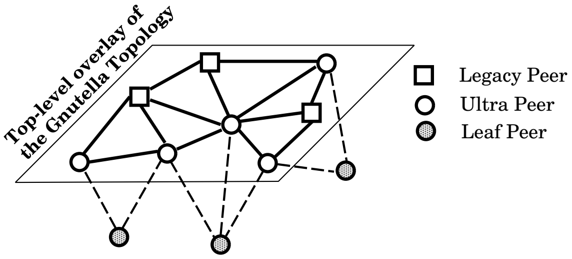


FIGURE 3.1. Gnutella's two-tier overlay topology

balance between ultrapeers and leaf peers, and *(ii)* a well-connected top-level overlay where each ultrapeer has a configured number of neighbors. To achieve these goals, participating peers collectively implement two mechanisms: First, a *promotion/demotion* mechanism that determines when a leaf should be promoted to become an ultrapeer and vice versa. Second, an *ultrapeer discovery mechanism* that enables either ultrapeers to find a neighbor or leaf peers to locate a parent in the top-level overlay with available open slots for neighbor or child peer, respectively.

The properties of the two-tier overlay in a widely-deployed P2P system depend not only on the portion of peers that support this feature but also on the coherency (or compatibility) of implementations (and configuration parameters) among participating peers. These properties can be further aggravated in open-source P2P applications since users can arbitrarily change their software. This raises the basic question of: *how can such a fluid two-tier overlay topology effectively accommodate such a rapid increase in peer population despite the heterogeneity of client software while maintaining a short pairwise distance among peers?*

This chapter, presents our investigation to answer the above question by empirically examining the long-term evolution of the Gnutella two-tier overlay topology during the last 15 months over which the user population has more than quadrupled. Using accurate snapshots of the Gnutella overlay, we characterize the following three angles of its long-term evolution: client, graph-related, and geographical properties of the overlay. We explore potential correlation between different observed characteristics and take the steps to identify some of the underlying causes.

The results presented in this chapter lead to two major findings: First, in response to the quick growth in user population, Gnutella successfully maintained its desirable graph properties by making modifications in the major client software releases and users contributed to this success by quickly upgrading to newer software releases. Second, we noticed a strong bias in the connectivity of peers towards other peers in the same region (continent). This observation was more outstanding in continents with smaller user populations and was maintained during the dramatic growth of the user base. The main contribution of this chapter is to illustrate the long-term evolution of a two-tier overlay in a widely-deployed P2P system while it has coped with a significant increase in user population. While it is extremely difficult to pinpoint the underlying causes of every observed characteristic in a large P2P system, this study sheds some light on how P2P overlays evolve.

In an earlier publication [131], my coauthors characterized graph-related properties of the Gnutella overlay topology across several snapshots (spanned over a few months) in order to provide representative results. This study complements mentioned earlier work by focusing on long-term trends in the two-tier overlay topology.

The rest of this chapter is organized as follows: In Section 3.1., we explain the importance of capturing accurate snapshots of a P2P system and briefly present my data collection methodology, my measurement tool and dataset. Section 3.2. presents the evolution of overlay properties in the Gnutella network.

3.1. Data Collection

To accurately characterize P2P overlay topologies, we need to capture *complete* and *accurate* snapshots. By “snapshot”, we mean a graph that captures all participating peers (as nodes) and the connections between them (as edges) at a single instant in time. The most common approach to capture a snapshot is to crawl the overlay. In practice, capturing accurate snapshots is challenging due to the large size and the dynamic nature of P2P systems. Because overlays change as the crawler operates, captured snapshots are inherently distorted where the degree of distortion is proportional to the crawling duration [128].

In this study, we use an efficient Gnutella crawler tool, namely *Cruiser* [127]. The measurement techniques developed in this tool improve the accuracy of captured snapshots by significantly increasing the crawling speed primarily through two mechanisms. First, it leverages the two-tier structure by contacting only ultrapeers. Since leaf peers connect only to ultrapeers, all of their topological information can be captured without contacting them directly. Second, Cruiser significantly increases the degree of concurrency in crawling by running on several machines and opening hundreds of simultaneous connections from each machine.

Cruiser can capture the Gnutella network with 2.2 million peers in around 8 minutes, or around 275 Kpeer/minute (by directly contacting 22 Kpeer/minute). This is orders of magnitude faster than the fastest previously reported crawler (2.5Kpeers/minute in [116]). Cruiser captures the following information from each peer it successfully contacts: *(i)* peer type (ultrapeer or leaf), *(ii)* brand and version of client, *(iii)* a list of the peer’s neighbors, and *(iv)* a list of an ultrapeer’s leaf nodes. Since the crawler does not directly contact leaf peers, I do not have information about their brand and versions.

3.1.1. Dataset

We have captured around 20,000 snapshots of the Gnutella network using Cruiser between Oct. 2004 and Jan. 2006 [1]. To minimize any possible error on the long-term analysis due to the time-of-day or day-of-week variations in overlay characteristics, we select 18 comparable snapshots that are taken around 3pm PDT on weekdays scattered during the 15-month measurement period [2].

3.2. Evolution of Overlay Properties

This section, presents the evolution of the two-tier overlay over a 15-month period. In the following subsections, we examine the evolution of three aspects of the Gnutella overlay topology: *(i)* the composition of participating clients, *(ii)* graph-related properties, and *(iii)* geographical properties.

3.2.1. Client Properties

Figure 3.2.a illustrates the growth in the population of Gnutella network during the past 15 months, and the breakdown of participating peers between the two levels of the overlay. This figure shows that the population has quadrupled during this period. The growth in population has been surprisingly linear with a noticeable dip over the 2004–2005 winter holiday season.

Now, we explore the different varieties of Gnutella clients in use and observe how users upgrade their software as new versions are released. Figure 3.2.b depicts the breakdown of ultrapeers across the major brands that implement Gnutella.

[1] Unfortunately, we did not capture any snapshots during May or June of 2004.

[2] While we do have a huge number of snapshots, the number of *comparable* snapshots is significantly smaller.

This figure shows that the two most popular implementations are LimeWire and BearShare. Overall, the ratio between LimeWire and BearShare has been fairly stable, with LimeWire making up 75–85% of ultrapeers, BearShare[3] making up 10–20%, and other brands making up 3–7%.

Gradual upgrading by users implies that different versions of each brand coexist at any point of time. P2P systems may need to evolve quickly in order to accommodate growing user demand. Otherwise, users may not observe acceptable performance and leave the system. This raises the following fundamental question: *"How rapidly and effectively can a widely-deployed P2P system evolve in order to cope with increasing user demand?"*

Since LimeWire clients make up an overwhelming majority of ultrapeers, we explored the breakdown among popular versions of LimeWire. Figure 3.2.c shows the percentage of LimeWire ultrapeers running each version, revealing that within 2 months of the release of a new version most LimeWire users are running it. This is illustrated by the way the market share of a version quickly increases from 0% to more than 50%, and only decreases when a new version appears. This behavior can be attributed to the automatic notification of new versions coupled with the simplicity of using the P2P system for distributing updates quickly. The quick upgrade by users also implies that new features rapidly become widespread throughout the system. Due to the rapid deployment of new versions, "flag days" are practical in P2P systems where new clients are configured to use a new, incompatible feature on a particular date.

[3] BearShare clients support more leaves per ultrapeer, and thus tend to have fewer ultrapeers. Therefore, while my results accurately represent the top-level overlay, they could potentially under-represent BearShare users.

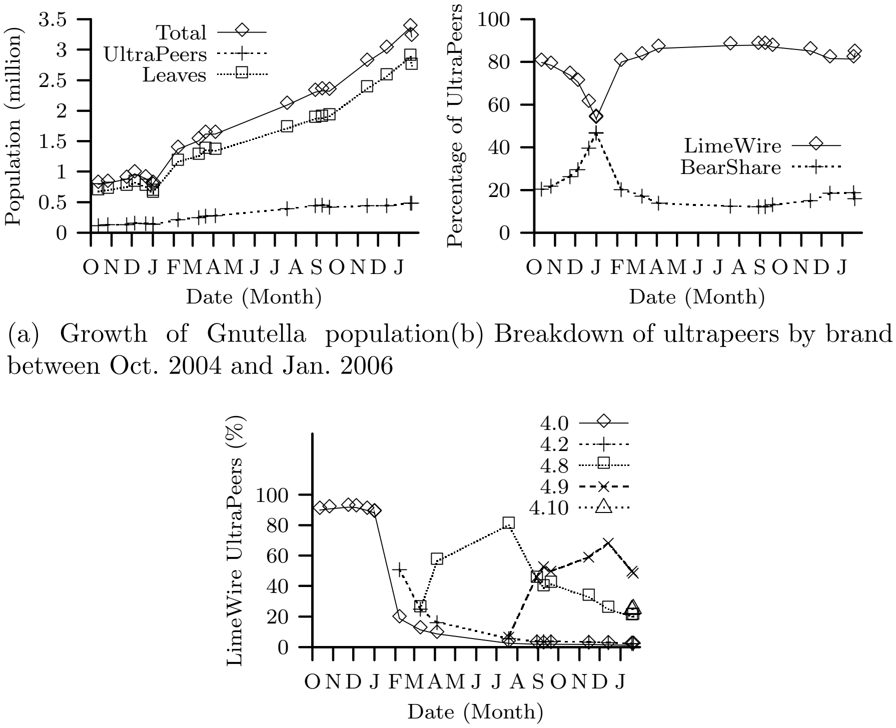
3.2.2. Graph-related Properties

We now turn our attention to the evolution of different graph-properties of the overlay topology.

3.2.2.1. Ultrapeer to Leaf Ratio

A key property of the two-tier overlay is the balance between the population of ultrapeers and leaves. We know that each ultrapeer attempts to maintain 30 leaf children, and each leaf tries to maintain 3 ultrapeer parents. Given the number of ultrapeers in the system, $|U|$, and the number of leaves, $|L|$, we can reason that there are $30 * |U|$ slots available for leaves, of which $3 * |L|$ are in use. If the ultrapeer-promotion mechanism is working well, and leaves can efficiently locate parents with open slots, then we would see few open slots (δ), *i.e.*, $(30 - \delta) * |U| = 3 * |L|$. For $\delta = 0$, fulfilling this equation yields a mix of 9% ultrapeers and 81% leaves. However, if δ is very small, this indicates that the system is working very hard to keep the balance perfectly despite constant churn in the system. To allow some flexibility, in practice the target percentage of ultrapeers is slightly more than this minimum of 9%, in order to provide some resiliency against dynamics.

Figure 3.4.a presents the change in the percentage of ultrapeers during the measurement period. As the population has grown, the percentage of ultrapeers have increased and reached two clear peaks (on Jan. and Sep. 2005), but has dropped back to the expected value (around 15%) in both cases. In Gnutella, leaf peers become ultrapeers only when they cannot locate a sufficient number of ultrapeers that can accept an additional leaf [122]. This increase in the percentage of ultrapeers illustrates the inability of leaves to locate available ultrapeers as the system has grown in size. However, the problem has been apparently addressed in the newer version of the client



(a) Growth of Gnutella population between Oct. 2004 and Jan. 2006

(b) Breakdown of ultrapeers by brand

(c) Breakdown of LimeWire ultrapeers by version

FIGURE 3.2. Evolution of client properties

which led to the drop in the percentage of ultrapeers. There seems to be a correlation between the drop in percentage of ultrapeers in Sep.–Oct. 2005 and the increase in popularity of LimeWire version 4.9, shown in Figure 3.2.c and discussed earlier.

3.2.2.2. Node Degree

To investigate changes in the connectivity of the overlay topology, we examine three different angles of the node degree distribution in the two-tier overlay: *(i)* for ultrapeers, the number of ultrapeer neighbors; *(ii)* for ultrapeers, the number of leaf children; and *(iii)* for leaves, the number of ultrapeer parents[4]. To show the

[4] We limit the range of node degree to 500 in these graphs. This range includes all but a small percentage of peers (<0.1%) with a higher degree.

evolution of the degree distribution over time, we have examined each angle of the degree distribution for all candidate snapshots. However, for clarity of the presented results, we show only four evenly spaced snapshots. The presented trends were similar across other snapshots except where noted.

In the absence of other factors, as the population grows, one expects the distribution to change proportionally across different degree values, *i.e.*, the ratio of peers with different degree would remain approximately constant. Figure 3.3.a shows the distribution of the number of top-level neighbors across ultrapeers for four snapshots in a log-log plot. All four distributions show a strong peak in the range of 20 to 30 neighbors, with a significant number of peers having less than 20 neighbors. Comparison of these snapshots reveals that the peak has dramatically grown, while the number of peers with fewer than 20 neighbors has increased only slightly rather than proportionally. This implies that despite the dramatic growth in the total population, ultrapeers with open slots for neighbors continue to quickly locate one another and form a well connected top-level overlay.

Figure 3.3.b shows the distribution of the number of leaf children across ultrapeers for four snapshots in a log-log plot. In all four snapshots, there are peaks at 30 and 45 children, corresponding to the maximums set in LimeWire and BearShare, respectively. However, unlike the number of neighbors, the peaks have not significantly increased over time. Instead, the dramatic increases have been in the number of ultrapeers with fewer children. This means that there are proportionally more ultrapeers with open slots for more children. This is the direct result of the unnecessary increase in the percentage of ultrapeers as illustrated by the two peaks in Figure 3.4.a. However, the increasing trend in the number of ultrapeers with open slots has reversed in the most recent snapshots as a result of drop in the percentage of ultrapeers during recent months. Note that the number of peers with fewer children has dropped between the two most recent snapshots in figure 3.3.b (*i.e.*, 7/19/05 and 1/20/06).

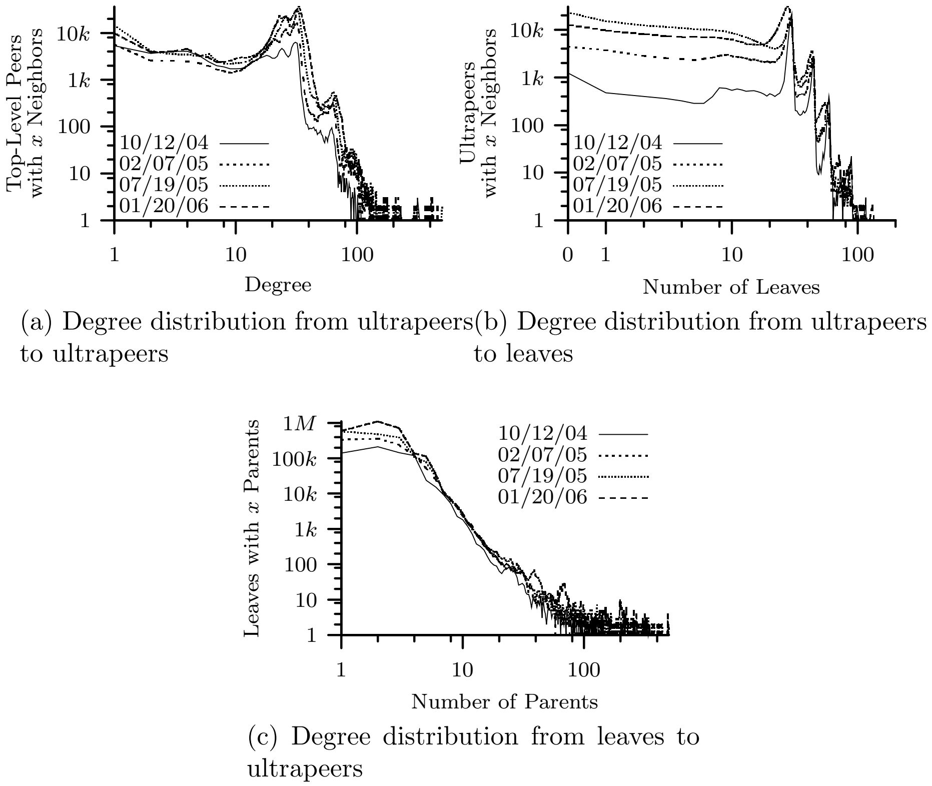
Figure 3.3.c shows the distribution of the number of ultrapeer parents among leaves in a log-log plot. In all snapshots, there is a peak at 1–3 parents, with many peers having slightly more parents. While the number of peers with 1–3 parents has proportionally increased with the population, the number of peers with more parents only exhibits a minor increase. This seems reasonable given the fact that both LimeWire and BearShare clients attempt to maintain 3 ultrapeer parents by default whereas peers with fewer parents are trying to find 3 parents. It also shows that the number of peers with more parents, presumably due to modified implementations, have not increased.

3.2.2.3. Clustering Coefficient

To examine the degree of clustering in the overlay topology, Figure 3.4.b depicts the evolution of the clustering coefficient during the measurement period. Comparing this figure with the population of ultrapeers (Figure 3.2.a) shows the clustering coefficient is inversely related to the population of ultrapeers. Since the degree distribution among ultrapeers is relatively fixed, as the number of ultrapeers increases, the top-level overlay becomes more sparse (*i.e.*, a smaller percentage of the possible edges exist), resulting in a lower clustering coefficient.

3.2.2.4. Pair-wise Distance

The distribution of pair-wise distances among pairs of peers is another interesting aspect of the overlay topology that determines the maximum useful scope for proper



(a) Degree distribution from ultrapeers to ultrapeers

(b) Degree distribution from ultrapeers to leaves

(c) Degree distribution from leaves to ultrapeers

FIGURE 3.3. Different angles of degree distribution

reachability in some search mechanisms. Figure 3.4.c depicts this distribution between *all* pairs of participating peers for three snapshots during the measurement period[5]. This figure illustrates that the significant growth in the population of peers has led to only a minor increase in the distances between peers. This is not surprising because of the logarithmic effect of population on the distances between peers in randomly connected graphs.

[5] Since the required processing for pair-wise distances is expensive ($O(n^2)$), we only conducted this analysis for these three snapshots.

3.2.2.5. Resiliency to Peer Departure

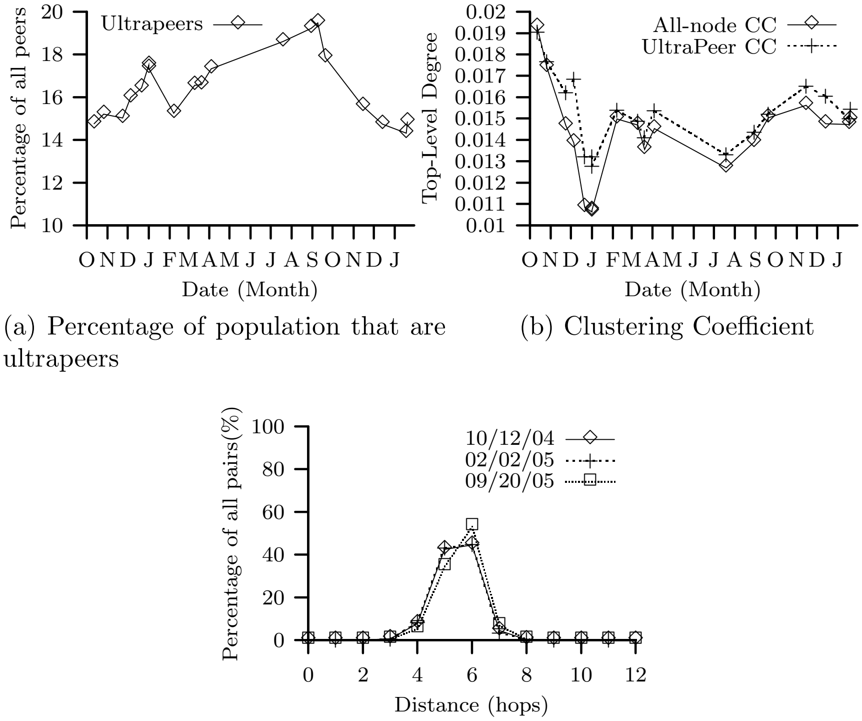
Finally, we examine the resiliency of the Gnutella overlay topology to both random and highest-degree node removal (or failure). Figure 3.5.a shows the percentage of ultrapeers that must be removed for the largest connected component to contain fewer than 50% of the remaining ultrapeers (*i.e.*, the overlay becomes severely fragmented). This figure shows that more than 90% of peers must be randomly removed from the overlay for it to become severely fragmented. Furthermore, the degree of resiliency has remained relatively constant during the past year. Resiliency to the removal of the highest-degree nodes is clearly worse than random node removal. Overall Gnutella is growing increasingly resilient to highest-degree removal. Since these results are normalized by total population, the actual number of removed ultrapeers has increased by a factor of 3 (i.e., $n * 50\%$ in Oct. 2004, $n * 3 * 60\%$ in Sep. 2005).

3.2.3. Geographical Properties

While neighbor selection is largely a random process in Gnutella, one key question is whether connectivity in the Gnutella overlay topology is geographically-aware. In other words, whether peers in a certain region are more likely to connect to other peers in their region.

3.2.3.1. Client Location

To characterize this property, first we examined the breakdown of ultrapeers across different regions and countries using GeoIP 1.3.14 from MaxMind, LLC. Figure 3.5.b shows the distribution of Gnutella clients across four regions, namely North America (NA), South America (SA), Europe (EU), and Asia (AS) that



(a) Percentage of population that are ultrapeers

(b) Clustering Coefficient

(c) Pairwise distance between pairs of peers

FIGURE 3.4. Evolution of graph properties

collectively make up 98.5% of the total ultrapeer population. This figure reveals that a majority of Gnutella ultrapeers are in North American (80%) with a significant fraction (13%) in Europe. Furthermore, the user population of different regions have grown proportionally over time. The distribution of user populations across different countries has also grown proportionally, except for China where client population has dropped significantly (94%). Clients in US, Canada, and UK make up 65%, 14%, and 5% of the total population, respectively[6]. The remaining countries made up less than 2% each, but make up 16% in total. Thus, while the Gnutella network is dominated

[6] These values are from the snapshot taken on 9/20/05 and are similar to the other values observed during the study period, as shown in Figure 3.5.b.

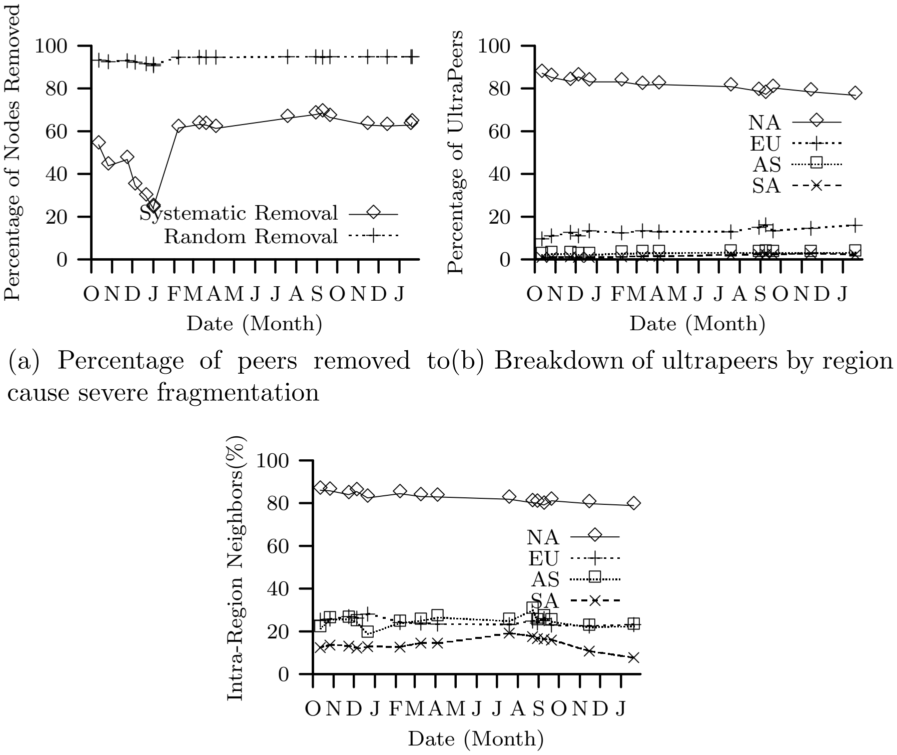
by predominately English-speaking countries, around one-fifth is composed of users from other countries[7].

3.2.3.2. Intra-Region Bias in Connectivity

For each one of the main four regions, Figure 3.5.c depicts the percentage of neighbors for all ultrapeers in a region that are located in the same region. If there is no bias towards intra-region connectivity, the percentage for each region should be the same as the percentage of the total population that are located in that region (Figure 3.5.b). Figure 3.5.c reveals that there is a strong bias towards intra-region connectivity, especially within smaller regions. More specifically, even though 13.3%, 2.8%, and 2.3% of the overall population are located in EU, AS and SA, more than 22.9%, 24.5%, and 16% of their neighbors are within the same region, respectively.

This biased intra-region connectivity occurs due to three reasons: First, LimeWire clients attempt to maintain at least one neighbor with the same locale setting [16], *i.e.*, at least one neighbor whose user speaks the same language. Second, when peers are attempting to establish more neighbors, they initiate connections to more peers than are actually needed and select the fastest responders, dropping any extras. This simple mechanism implicitly leads to bias in connectivity within each region. Third, because users in the same region tend to arrive at around the same time of day, their clients tend to be looking for neighbors at the same time and are more likely to find one another. Clearly, one could determine the potential for such a biased connectivity by examining the source code of various implementations. However, my results quantify the degree of such bias in practice.

[7] We noticed that, the population of North American and European clients peak at around 7pm and 11am PDT with 86% and 24%, respectively. This figure indicates that the 3pm snapshots capture roughly average daily population, *i.e.*, not at any of the peaks.



(a) Percentage of peers removed to cause severe fragmentation

(b) Breakdown of ultrapeers by region

(c) Percentage of neighbors in the same region

FIGURE 3.5. Resiliency and geographical properties

This intra-region biased connectivity in the overlay topology implies that users searching for content are more likely to locate content among other peers in the same region with the same language and culture. Furthermore, response time to queries will also be faster since geographical distance is a good first-order estimator of network latency.

3.3. Summary

In this chapter, we explored long-term trends in properties of the overlay topology in the popular Gnutella P2P file-sharing system. In particular, we illustrated how the two-tier overlay topology has evolved in order to accommodate dramatic changes in the scale of the user population during the 15 month measurement period. The rapid rate of software updates by participating users enabled developers to effectively modify their software to cope with this moving target and maintain a two-tier overlay with desired properties. We have explored potential correlations between the evolution of overlay properties and the popularity of different versions of major client releases. Finally, we illustrated the intra-region bias in the connectivity among peers.

The main two findings of this chapter are the following: First, the quick growth in user population begun to push the Gnutella overlay past its limits. However, the developers quickly responded by making modifications in major Gnutella clients (LimeWire and Bearshare) and the users quickly adopted the new releases and thus the desirable properties of the overlay were maintained despite the dramatic growth in user population.

Specifically;

- Ultrapeer to leaf ratio target is a little over 9% for good performance. During the study, the ratio became unbalanced reaching peaks of 18% and 20%. However, in both cases, the Gnutella software upgrades were able to respond, bringing the ratio to an acceptable level of about 15%.

- Our study of client-based properties of the Gnutella overlay showed a dominance of one implementation (Limewire) by 75-85% of ultrapeers, with the second implementation (BearShare) making up 10-20% of ultrapeers. Clients quickly adopt new upgrades as they are released, providing rapid adaptation to any instability that results from rapid growth of the user population.

- Our study of graph-related properties of the Gnutella overlay (ultrapeer node degree, degree from ultrapeers to leaves, and leaves to ultrapeers) shows that similar patterns occur over multiple snapshots in time. Peak values occur at certain specific values in all snapshots, but all are consistent with a system that reacts successfully to increasing numbers of peers, *i.e.*, new releases of Limewire and BearShare bring the P2P network into reasonable balance.

Our study of other graph-related properties (clustering co-efficient, pair-wise distances, and resiliency to peer departures) all exhibit characteristics that are; *(i)* consistent with expectations, and *(ii)* indicative of stability *i.e.*, these metrics remain in ranges that are appropriate for effective performance for Gnutella's file-sharing needs.

Second, in Gnutella overlay, despite the general randomness, peers show a meaningful bias in connecting to other peers in the same continent, specially in continents with smaller user populations. This connectivity bias has not changed during the dramatic growth in user population. Our study of geographic properties of the Gnutella P2P overlay suggests that Gnutella is much more popular in English-speaking countries and there is a strong bias toward intra-region connectivity. Although this is not a surprising result, we are the first to quantify this pattern.

The key contributions of our work include the rigorous measurement of graph theoretic and geographic characteristics of the Gnutella overlay network. Through use of the Cruiser crawler, we have captured much more accurate measurements than previously reported. Our measurement study has also taken more snapshots than other work because of the speed of Cruiser. These measurements will serve as useful baseline measurements for future studies of P2P overlay characteristics.

Our measurements over a two-year period demonstrate the ability of Gnutella software (most notably Limewire) to provide mechanisms to adapt to rapid growth in user populations through fast adoption of Limewire releases. Coupled with the stability in the geographic properties of the P2P overlay, this analysis provides information that will be very useful in our investigation of the impact of the overlay on the underlay network.

The measurement technique used in this chapter was based on taking full snapshots of a live P2P overlay in a relatively short time slot. Although this method was shown to be effective for Gnutella at the time the study was performed, we should note that this technique becomes very challenging as the network becomes more and more populated. For instance, if a P2P network has 10 times the maximum number of peers we observed in this study (35 million peers), taking a full snapshot will take about ten times longer, making the snapshots likely to be distorted as we will observe in Section IV. In the next chapter, we will introduce an efficient sampling technique that will significantly reduce the measurement time in order to improve the accuracy of the measurement for even larger P2P networks.

CHAPTER IV

LARGE SCALE OVERLAYS: SAMPLING

Most of the content from this chapter has been adopted from my previously published paper [103] co-authored with Mojtaba Torkjazi, Prof. Reza Rejaie, Dr. Nick Duffield, Dr. Walter Willinger, and Dr. Daniel Stutzbach. The experimental work is mine with some assistance from Mr. Torkjazi and the text has been contributed by myself and the co-authors.

During the past few years, unstructured Peer-to-Peer (P2P) systems such as Gnutella and BitTorrent have become very popular and have significantly contributed to the total traffic over the Internet. This has motivated researchers to characterize the basic properties of these systems through measurement. Such characterizations can be leveraged to address several key issues about these systems including: *(i)* understanding the properties and dynamics of these systems, and use these findings to improve their performance and scalability, and *(ii)* assessing the impact of these systems on the Internet.

To characterize unstructured P2P systems, one needs to capture accurate "snapshots" of the connectivity structure. Examining individual snapshots reveals the connectivity structure at a particular point of time whereas comparing consecutive snapshots over time illustrates the evolution of the connectivity structure. Such snapshots are typically captured by a crawler that queries a set of known nodes to learn about their neighbors and progressively discovers the connectivity structure. Capturing accurate snapshots of the connectivity structure for large-scale unstructured overlays is challenging because such systems may significantly evolve during the time required to capture a full snapshot. Therefore, captured snapshots are likely to be *distorted* and this could significantly degrade the accuracy of any results derived from such snapshots. In [130], my co-authors have shown that commonly used sampling techniques in prior empirical studies on P2P systems (*e.g.*, [115]) can easily lead to significant bias towards short-lived or high degree peers due to the dynamics of peer participation or the heterogeneity of peer degrees, respectively. *Graph sampling* (*e.g.*, [130]) is a natural approach to tackle this problem and often occurs in two steps. First, a crawler explores parts of the structure and selects a (random) subset of discovered nodes as samples. Second, the desired property of sampled nodes is measured to yield an estimate of the distribution of that node property across the entire population.

This chapter presents *Respondent-Driven Sampling (RDS)* as a promising technique for sampling unstructured P2P overlays. This allows one to accurately estimate the distribution of a desired peer property without capturing the entire overlay structure. RDS is a variant of snowball sampling that has been proposed and used in the social sciences to characterize hidden population in a society [62, 114]. We apply the RDS technique to unstructured P2P network and evaluate its performance over a wide range of static and dynamic graphs as well as a widely deployed P2P system. Throughout our evaluation, we compare and contrast the performance of the RDS technique with another sampling technique, namely *Metropolized Random Walk (MRW)*, that we developed in our earlier work [130].

The presented results illustrate three main findings: First, the performance of RDS is equal or better than that of MRW in all examined cases. The advantage in performance and accuracy is most outstanding in cases where the overlay structure features a highly skewed node degrees while the node clustering coefficients are also highly skewed. Second, both sampling techniques exhibit acceptable performance in accurately estimating peer properties over dynamic unstructured overlays according to our dynamic simulations as well as empirical evaluation. Third, we observed lower efficiency and accuracy from both techniques in empirical evaluation compared to the simulations results. We believe this is due to the fact that we are unable to obtain perfect reference snapshots in real world experiments.

The rest of this chapter is organized as follows: Section 4.1. presents an overview of both the RDS and MRW techniques, and sketches our evaluation methodology. We examine both techniques over variety of static and dynamic graphs in Section 4.2. and 4.3., respectively. Section 4.4. presents the empirical evaluation of the two sampling techniques over Gnutella network.

4.1. Graph Sampling Techniques

An unstructured overlay can be represented as an evolving undirected graph G, with vertices V and edges E. The vertices and edges of G represent the peers and pairwise connections between them, respectively. An accurate snapshot of the full graph (the overlay) is not available, however we can query any known peer for a list of adjacent peers in order to progressively discover portions of the overlay. Some fraction of discovered peers are selected as samples and the distribution of the desired peer property (number of neighbors, number of files, access link bandwidth or session time) among the samples provides an estimate for that property among all peers. The efficiency of sampling can be quantified by the ratio of sampled peers to the total number of peers queried. To provide an accurate estimate, sampled peers should be selected uniformly at random. This is challenging because the overlay topology and peer dynamics introduce bias towards discovery and thus selection of peers with large degrees and short session times, respectively [130].

Random walk is a promising technique for sampling. In an ordinary random walk, the sampler begins at a node, x, and chooses a new node, y, uniformly at random from x's neighbors. The walk transitions to the neighbor and then chooses a new node from y's neighbors. Formally, the ordinary random walk has a transition function, $P(x, y)$, defined as follows:

$$P(x,y) = \begin{cases} \frac{1}{\text{degree}(x)} & y \text{ is a neighbor of x,} \\ 0 & \text{otherwise} \end{cases}$$

The *stationary distribution*, $\pi(x)$, of the walk defines the probability of being at any particular node x. For an ordinary random walk, graph theory [78] proves $\pi(x) \propto \text{degree}(x)$. That is, the fraction of time spent at a node is directly proportional to the node's degree. Thus, the ordinary random walk is inherently biased towards nodes with higher degree.

4.1.1. Respondent Driven Sampling

Respondent Driven Sampling (RDS) is a development of Snowball Sampling (SBS)[62], a group of related sampling techniques proposed in the social sciences to sample hidden populations. Salganik [114] defines a population as "hidden" when there is no central directory of all population members, such that samples may only be gathered through iterative referrals from existing samples.

RDS is a variant of SBS [62], which forms asymptotically unbiased estimators by appropriate re-weighting of estimators to take account of topological biases [114]. The special case where each respondent recruits only one individual maps exactly onto the case of a random walk on a graph. This in turn can be recast as a Monte Carlo Markov Chain (MCMC) problem [51] The problem of estimating peer properties in unstructured overlays is analogous to the sampling of hidden population in the social sciences. We wish to estimate the distribution of a node property X; specifically, consider any partition $\{R_1, \ldots, R_m\}$ of the range of possible values of X. We partition the node set V accordingly into groups of nodes $\{V_1, \ldots, V_m\}$, *i.e.*, $V_i = \{v \in V : X(v) \in R_i\}$. A simple example is when X is positive integer value and we group by value: $V_i = \{v \in V : X(v) = i\}$.

The RDS approach is to estimate the proportion p_i of nodes that are in group i from observed node degree and group memberships of nodes traversed in the random walk. Specifically, consider the n-step walk that visits the set of nodes $T = \{t_1, t_2, \ldots, t_n\}$ where individual nodes may be visited more than once. Let $T_i = T \cap V_i$ denote the visited nodes that lie in group i. For any node property X, the Hansen-Hurwitz [58] estimator $\hat{S}(X) := n^{-1} \sum_{v \in T} \frac{X(v)}{\pi(v)}$ is an unbiased and consistent estimator of the sum $S(X) := \sum_{v \in V} X(v)$ when T is drawn from a stationary random walk, *i.e.*, one that evolves from an initial node that is randomly selected according to the stationary distribution. Consider two special cases. When $X = I_{V_i}$ is the indicator of a node being in group i, *i.e.*, $I_{V_i}(v) = 1$ if $v \in V_i$ and 0 otherwise, then $\hat{S}(I_{V_i})$ estimates the total number of nodes in V_i. When $X = 1$ then $\hat{S}(1)$ estimates the total number of nodes $|V|$ in the graph. Thus we can estimate the proportion p_i by

$$\hat{p}_i = \frac{\hat{S}(I_{V_i})}{\hat{S}(1)} = \frac{\sum_{v \in T_i} \frac{1}{\text{degree}(v)}}{\sum_{u \in T} \frac{1}{\text{degree}(u)}}$$

where degree(v) is the degree of the node v. $\hat{p}_i$ is consistent—it converges to the true value p_i—as the number n of visited nodes grows. The RDS estimator can be recognized as an importance sampling estimator weighted by the stationary distribution π, applied to the MCMC of the random walk on the vertex set V.

4.1.2. Metropolized Random Walk

Our earlier work [130] evaluates the use of Metropolized Random Walks (MRW) for gathering unbiased samples from unstructured P2P networks. The Metropolis–Hastings technique [30, 60, 90] provides a way to alter the next-hop selection to produce any desired stationary distribution, $\pi(x)$. In [130], we choose the next-hop appropriately to produce the uniform distribution, $\pi(x) = \frac{1}{|V|}$, as follows:

$$Q(x,y) = \begin{cases} P(x,y)\min\left(\frac{\text{degree}(x)}{\text{degree}(y)}, 1\right) & \text{if } x \neq y, \\ 1 - \sum_{z \neq x} Q(x,z) & \text{if } x = y \end{cases}$$

Essentially, the walk tentatively selects a neighbor of x uniformly at random $(P(x,y))$ and then accept the transition randomly with probability $\min\left(\frac{\text{degree}(x)}{\text{degree}(y)}, 1\right)$. Otherwise $(1 - \sum_{z \neq x} Q(x,z))$, the walk remains at the current node, effectively taking a self-edge. Put simply, the bias toward higher degree nodes is removed by reducing the probability of transitioning to higher degree nodes at each step.

We note that RDS is complementary to the MRW approach in the following way. In MRW, we seek to modify the random walk in order to have an equal probability of visiting each node and hence derive unbiased estimates. In RDS, the walk is unmodified; however, we reweight the sampled values to obtain an unbiased estimate of the group proportions p_i.

4.1.3. Evaluation Methodology

To evaluate the RDS technique, first we simulate the sampling techniques over a wide range of static and dynamic graphs where the accurate distribution of the sampled property (ground truth) is known. Simulation over synthetic graphs not only offers an opportunity for accurate evaluation of the sampling techniques, but also allows us to identify the separate effects of graph properties and graph dynamics on the accuracy and efficiency of these techniques. Second, we empirically evaluate both techniques over Gnutella P2P overlay.

Performance Metric: To quantify the accuracy of a sampling technique in each scenario, we compare the sampled and true distributions of a desired peer property using the Kolmogorov-Smirnov (KS) statistic, D. If we plot the estimated and true CDFs of a desired property, D is the maximum vertical distance between the plots of the two functions with a range of $[0, 1]$. For example, a value of $D \leq 0.01$ corresponds to no more than a one percentage point difference between CDFs and is excellent for most measurement purposes. Due to the limited space, we only present a subset of our results that illustrate our main findings in Sections 4.2. and 4.3.. Complete results are available in the related technical report [101].

4.2. Evaluation over Static Graphs

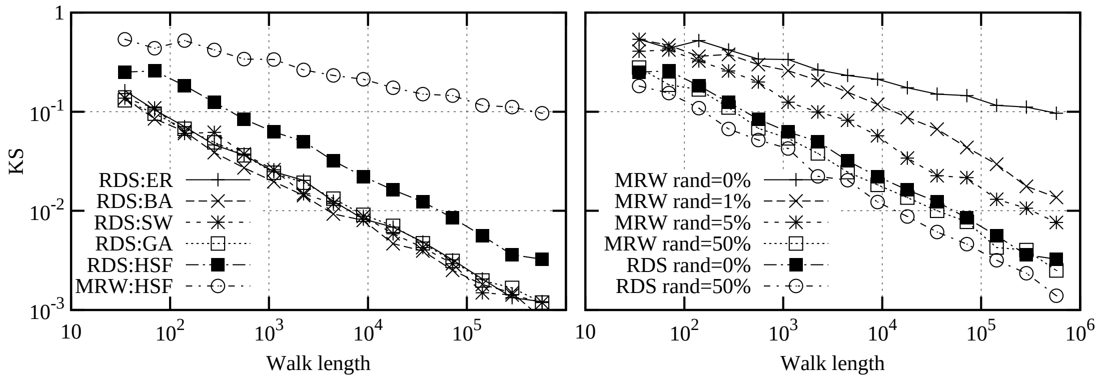
In this section, we examine how the connectivity structure of a graph affects the accuracy and efficiency of the RDS and MRW sampling techniques using the following candidate graph types: *(i) Erdös-Rényi Random graphs* (ER) [17], *(ii) Small-world graphs* (SW) [135], *(iii) Barabási-Albert graphs* (BA) [12]: Scale-free graphs of the preferential attachment-type, *(iv) Hierarchical Scale-Free graphs* (HSF) [13]: A class of (deterministic) graphs generated by an iterative algorithm to produce heterogeneous node degree and heterogeneous node clustering coefficients. More specifically, node degree distribution follows power-law while clustering coefficients at individual nodes is inversely proportional to node degree, independent of graph size. *(v) Gnutella graphs (GA)*: A snapshots of the Gnutella ultrapeer topology, captured on 05/15/2008 using *cruiser* [130].

Figure 4.1.a shows the KS error for the degree distribution from samples collected by the RDS and MRW techniques as a function of the number of samples over the different graph types. To make the results comparable, the number of vertices ($|V|$ = 390,625) and edges ($|E|$ = 1,769,110) are similar across the different graph types. Figure 4.1.a illustrates the following two important points. First, the accuracy of the RDS technique rapidly improves with the number of samples. The rate of improvement in accuracy across all graph types (*i.e.*, slope of the line) is similar. The overall accuracy of the MRW technique follows a trend similar to RDS for all graphs (except the HSF graph) but on average slightly ($\approx 2 * 10^{-3}$) lower than the RDS technique. Given this similarity, the results for MRW are not shown in Figure 4.1.a except for the HSF graph. For the HSF graph, MRW sampling not only exhibits a significantly lower accuracy compared to the other graph types, but the rate of improvement in accuracy with the sample size (*i.e.*, slope) is much worse. Second, for a given number of samples, while both techniques exhibit a lower accuracy for the HSF graph, the impact on the MRW technique is significantly more pronounced, *i.e.*, the rate of improvement in accuracy with the sample size (*i.e.*, slope) for the MRW technique is much worse than RDS.

Focusing on the HSF graph, the reported differences in the accuracy of RDS and MRW and their observed lower performance can be attributed to the following phenomenon. In HSF graphs, at each level of their hierarchical structure, there are groups of well inter-connected low degree nodes which form pronounced *clusters*. The only way for a random walker to leave these clusters is via a much higher degree node that resides outside these clusters, *i.e.*, the walker has to traverse an edge from a low degree node within such a cluster to a much higher degree node outside this cluster. As described in Section 4.1., for the MRW technique, the probability of moving along such an edge is proportional to the ratio of the (low) degree of the node within the cluster to the (very high) degree of the node outside this cluster which is very small. Therefore, when an MRW walker ends up in one of these clusters, it keeps collecting samples from low degree nodes within these clusters for a disproportionally long time. This in turn degrades the accuracy of sampling especially among high degree nodes. The impact of clusters on the RDS technique is significantly lower because the probability of selecting the next node in RDS does not depend on node degree.

Figure 4.1.b shows the accuracy of the MRW sampling technique over the same HSF graph when 0%, 1%, 5%, and 50% of its edges are randomly shuffled (*i.e.*, rewired) while preserving the degree of individual nodes. Increasing the percentage of randomly shuffled edges gradually removes the explicit hierarchical structure of HSF graphs and enforces a more homogeneous clustering behavior across the graph structure as compared to the original HSF graph. For comparison, we also present the results for the RDS technique over the graphs when 0% and 50% of edges are shuffled. The figure demonstrates that even a small percentage of shuffled edges dramatically improves the accuracy of the MRW technique.

These results suggest that the main reason for the degraded performance of the two sampling techniques over HSF graphs is a combination of highly skewed node degrees and highly skewed node clustering coefficients.



(a) Efficiency of RDS & MRW (b) HSF graphs with edge shuffling

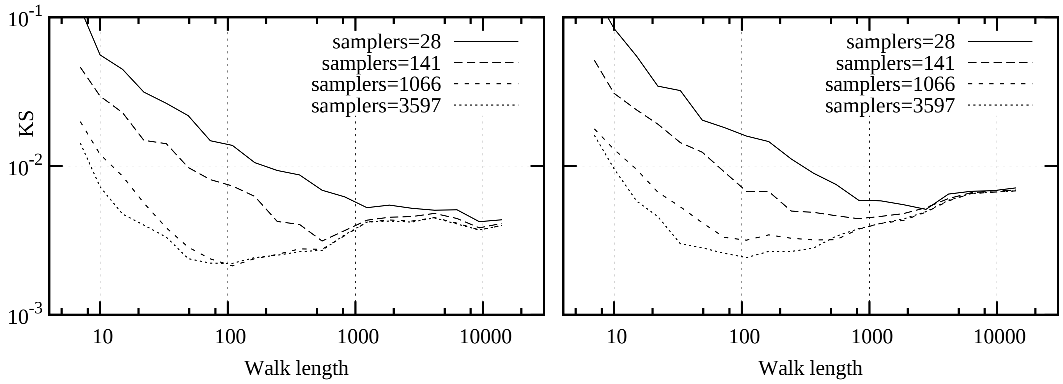
FIGURE 4.1. Efficiency of RDS and MRW over different graph types, estimating degree distribution

4.3. Evaluation over Dynamic Graphs

In this section, we use our session-level simulator [129], called *psim*, to examine the behavior of the RDS technique over dynamic graphs. *psim* simulates peer arrivals, departures, pairwise latencies, per discovery and neighbor connections. The latencies between peers are randomly selected from the King data set [56]. Peers use the following popular bootstrapping mechanisms for peer discovery [130]: *Oracle, FIFO, HeartBeat* and *History.* Individual peers try to maintain the number of their connections (*i.e.*, their degree) between a given minimum ($MinDeg$) and maximum ($MaxDeg$) degree. When the number of connections for a peer drops below $MinDeg$, it uses the discovery mechanism to establish additional connections and reach $MinDeg$. A peer neither accepts nor initiates any new connections once its degree reaches $MaxDeg$. To query a peer for a list of neighbors, the sampling node must establish a TCP connection, submit its query, and receive a response. *psim* simulates churn by controlling the distributions of peer inter-arrival intervals and peer session lengths. New peers arrive according to a Poisson process, where the mean peer arrival rate combined with the session length distribution yield a desired mean population size in steady state. We use the following models for session length distribution that we derived in our earlier empirical study of churn [129] in P2P networks: Weibull, Pareto and Exponential. We run each simulation for a warm-up period until it reaches steady state with 100,000 concurrent peers before gathering samples.

Impact of Parallel Sampling: A desired number of samples from a dynamic overlay can be collected by a number of parallel (RDS or MRW) walkers that start from the same nodes. Increasing the number of parallel samplers has two conflicting effects and thus introduces an interesting tradeoff. Increasing the number of parallel walkers, reduces the required walk length to collect a desired number of samples. This in turn decreases the time to collect the samples and thus reduces the error that occurs due to the evolution (*i.e.*, churn) in the overlay. However, increasing the number of samplers leads to redundant sampling of nodes around the starting point and degrades sampling accuracy. Figure 4.2. demonstrates this tradeoff and depicts the accuracy of the RDS and MRW techniques as a function of walk length for different number of parallel samplers.

Clearly, the accuracy of the RDS and MRW techniques in estimating a peer property is not affected by overlay dynamics if the desired peer property does not interact with the walk. Therefore, to evaluate these techniques over dynamic graphs, we only consider the following peer properties that may interact with the walk: *(i) Node Degree (DEG)*: The degree of an individual node in the graph determines the probability that a node is visited. *(ii) Session length or Uptime (UT)*: The dynamics of peer participation drives the evolution of the graph with time and affect the probability of visit for individual peers. *(iii) Query latency (RTT)*: In a dynamic



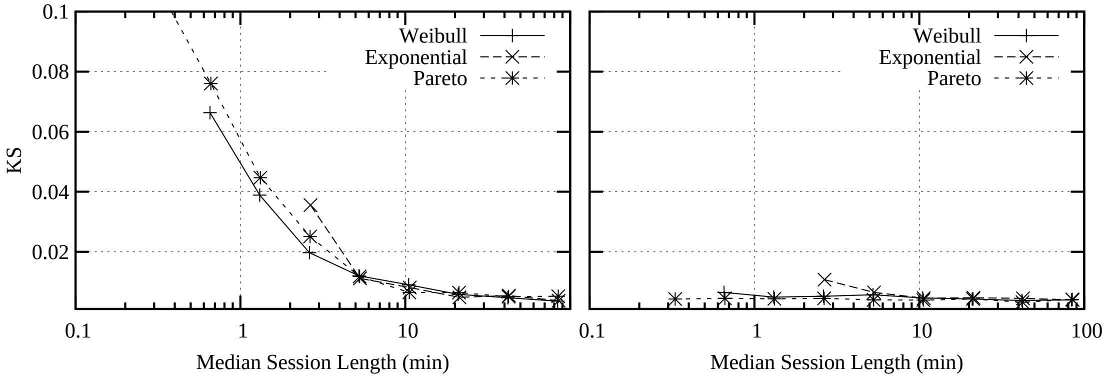
(a) Sampling Tech:RDS (b) Sampling Tech:MRW

FIGURE 4.2. Effect of number of samplers and walk length in sampling node degree. Churn Model: Weibull with $k = 0.59$, median sess. len.=21min, MinDeg=30

overlay, each step requires querying a peer. Since the query latency for individual peers depends on their relative round-trip time, this could lead to a bias correlated with the query latency.

4.3.1. Effect of Churn

Figure 4.3.a depicts the accuracy of the RDS technique in estimating the distribution of node degree as a function of median session length (*i.e.*, churn rate) for different churn models. The results for sampling session length is very similar to Figure 4.3.a and thus is not shown. Figure 4.3.a shows that the median session length is the primary factor that affects the accuracy of sampling techniques. To explain this behavior, we note that the median session length is a rough measure of the level of overlay dynamics. When the churn rate is high (*i.e.*, median session length is less than 5 minutes), the overlay significantly evolves during the sampling period which in turn leads to larger error. Earlier empirical studies suggest that the median session length in actual P2P systems is rarely below 10 minutes for which the sampling error is below 0.01. Figure 4.3.b presents the accuracy of the RDS technique in estimating



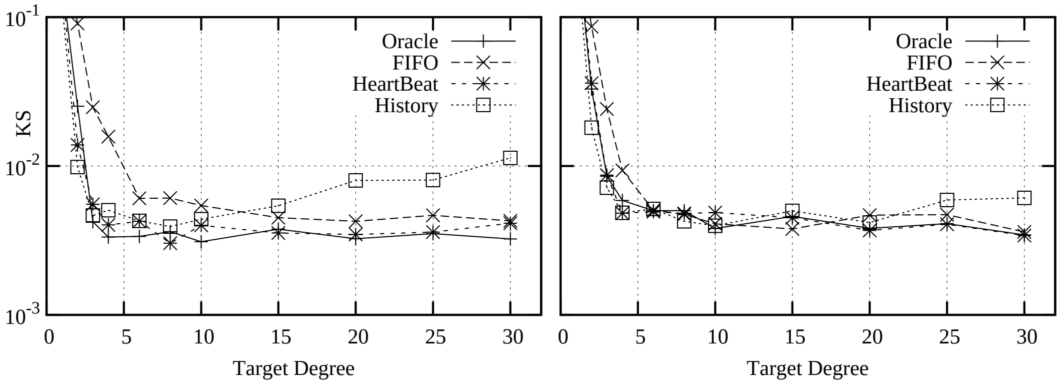
(a) Tech:RDS Prop:Deg (b) Tech:RDS Prop:RTT

FIGURE 4.3. Sampling error as a function of *median session length* for two peer properties using different churn models. Bootstrapping=FIFO, MinDeg=30, sampled by 1066 parallel samplers, each taking 49 hops

the distribution of *query latency*. Since the query latency between pairs of nodes are selected from the fixed King data set, its distribution among samples is less sensitive to the dynamics of peer participation.

4.3.2. Effect of Target Node Degree

Figure 4.4.a presents the accuracy of the RDS sampling technique in estimating the distribution of node degree as a function of minimum node degree ($MinDeg$). This figure reveals that when $MinDeg$ is larger than a threshold of about five, the accuracy does not change with the the minimum node degree (except for the *History* bootstrapping mechanism). Figure 4.4.b shows that the accuracy of the RDS technique in estimating query latency follows a similar pattern. The rapid degradation of accuracy for lower node degrees is mainly due to the fragmentation of the overlay which makes some parts of the graph inaccessible to the random walkers. In real P2P systems, such a fragmentation does not occur since the peer degree is often larger than five.



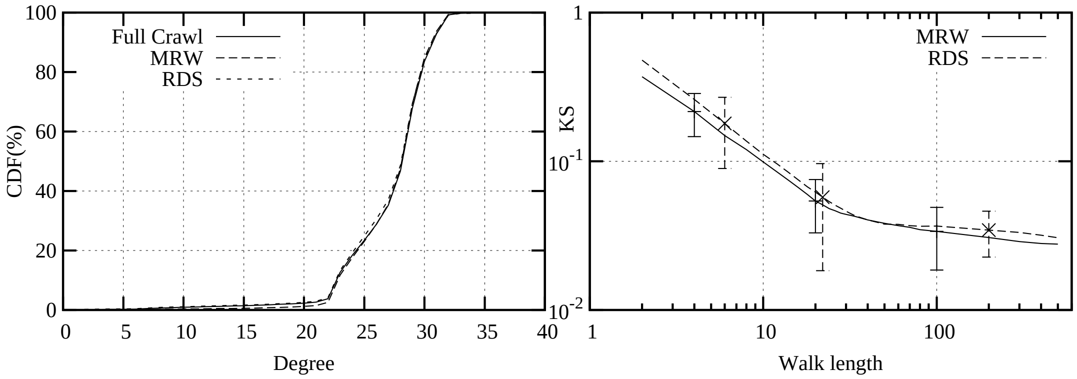
(a) Tech:RDS Prop:Deg (b) Tech:RDS Prop:RTT

FIGURE 4.4. Sampling error as a function of *minimum degree* for two peer properties using different bootstrapping mechanisms. Churn Model: Weibull $k = 0.59$, Med. sess. len.=21min, 1066 parallel samplers, each taking 49 hops

To explain the abnormal behavior of the *History* bootstrapping mechanism in Figure 4.4., we note that in this mechanism each peer relies on the list of its neighbors in previous sessions. This leads to a number of isolated peers since all their neighbors from previous sessions have departed. While the number of isolated peers is not large at any given time, and they eventually get connected to the overlay by contacting bootstrapping node, their extended isolation time have an impact on the reachability of these nodes and thus on the accuracy of sampling techniques.

4.4. Evaluation over Gnutella

To empirically evaluate our sampling techniques, we use them to estimate properties of ultrapeers in the Gnutella network on sampling peer properties of Gnutella, a well-known and popular P2P system. We incorporate both sampling techniques into our sampling tool called *ion-sampler*[130]. We concurrently start 1000 RDS and 1000 MRW samplers, where each sampler takes a 500-step walk to sample the degree of Gnutella ultrapeers in the Gnutella network to collect samples



(a) Degree distribution

(b) Node degree sampling error

FIGURE 4.5. Results of sampling experiment over Gnutella

of node degree. At the same time, we use *cruiser* [132] to collect complete back-to-back snapshots of the top-level Gnutella overlay roughly every seven minutes.

Figure 4.5.a presents the distribution of node degree from collected samples by the RDS and MRW techniques as well as full snapshots collected by the crawler. Figure 4.5.a shows that all three distributions of node degree are almost indistinguishable, *i.e.*, both sampling techniques exhibit similar performance. To further investigate the variability of observed accuracy for the sampling techniques, we repeat each sampling experiment with different walk length for six times. Figure 4.5.b presents the average KS error and associated error bars as a function of walk length for both sampling techniques. Figure 4.5.b indicates that increasing the walk length beyond about 30 hops quickly decreases the KS error because of the larger number of collected samples. However, the rate (*i.e.*, slope) of improvement in accuracy is diminishing beyond a certain walk length. Collecting more samples through longer walks does not improve the fidelity of samples due to major changes in the system during the sampling period.

4.5. Summary

In this chapter, we presented RDS as a powerful technique for sampling unstructured P2P networks. While RDS has been developed in the social sciences for sampling static graphs, we adopted this technique to the networking domain and explored its applicability in the context of dynamic connectivity structures. Through simulations involving a variety of synthetically generated static and dynamic graphs and experiments over the Gnutella network, we examined the performance of the RDS technique and compared its performance with another graph sampling technique, namely, MRW. Our study demonstrates how the connectivity structure among nodes and its dynamics affect the accuracy of both sampling techniques. We showed that RDS generally performs as good or better than MRW. In particular, RDS achieves a significantly better performance than MRW when the overlay structure exhibits a combination of highly skewed node degrees and highly skewed node clustering coefficients.

Our experiments on a variety of synthetically generated static graphs show that RDS and MRW perform equally well for most graph types including ER, BA, SW, and GA in which sampling accuracy is roughly inversely proportional to the square root of the sample size (walk length). For HSF graph, we observe that RDS performs much better than MRW. For instance comparing both techniques at their largest experimented sample size (nearly 1 million samples), MRW and RDS reach best sampling errors of roughly 10^{-1} and 3×10^{-3}, respectively.

We also examined the performance of RDS via dynamic simulation in presence of churn. We find that increasing churn level beyond a certain point (*i.e.*, when the median session length is less than 5 minutes), the overlay changes significantly during the sampling time and therefore the sampling accuracy is largely degraded. Also, when the set the target node degree (in overlay construction) below a threshold of about 5, the sampling accuracy is degraded significantly. We believe this is due to major fragmentation that occurs in the dynamic graph when the node degree is small.

We test RDS in a real world measurement by sampling the Gnutella network and we find that RDS and MRW perform equally well in sampling Gnutella reaching a sampling error of about 3×10^{-2}.

In the framework of measurement studies of P2P systems, in the past two chapters we focused on the P2P overlay and how to capture overlay/node properties using full crawling or sampling. Another aspect of P2P systems which is the most important from the user's perspective, is the performance experienced by the user. In the next chapter, we focus on P2P performance evaluation using a case study of *BitTorrent*.

CHAPTER V

P2P PERFORMANCE EVALUATION: BITTORRENT CASE STUDY

Most of the content of this chapter has been adopted from my previously published paper [105] co-authored with Prof. Reza Rejaie. The experimental work and analysis is entirely mine and the text has been contributed by myself and my co-author.

During recent years, the Internet has witnessed a rapid increase in the popularity of BitTorrent and therefore its contribution in network traffic. BitTorrent is a peer-to-peer (P2P) content distribution mechanism that enables a single node to provide its static content to a large number of peers without requiring a large access link bandwidth. BitTorrent incorporates swarming content delivery to effectively utilize the outgoing bandwidth of participating peers and thus achieve scalability. The scalability of BitTorrent along with the ease of deployment has led to its increasing popularity over the Internet. This in turn has motivated researchers to examine the performance of BitTorrent using different techniques including modeling ([98]), simulations ([15, 98]), and in particular measurement ([66, 57, 71]).

One key aspect of performance in BitTorrent is the download rate that is achieved by participating peers. It is often stated (by users and developers) that BitTorrent provides a good performance to individual peers, *i.e.*, users can effectively utilize their available (or configured) incoming access link bandwidth. However, capturing a "representative" value of observed peer performance in practice is a non-trivial task. A typical measurement approach to study peer-level performance is to use one (or multiple) instrumented BitTorrent client(s) that participate in an existing torrent and download content [66, 75]. While this approach provides detailed information about the observed performance by a few instrumented peers, it is unclear whether its findings properly represent observed behavior by the entire population, *i.e.*, the results may not be *representative*. Intuitively, the observed performance by individual peers in a torrent could depend on their peer-level properties (*e.g.*, outgoing access link bandwidth) or group-level properties (*e.g.*, group population, content availability or churn). This implies that results of a measurement study using "instrumented client" could easily depend on time of measurement, location of instrumented clients or properties of the torrent and thus they are not representative. In essence, to investigate the peer-level performance in BitTorrent, the following two important and related questions should be addressed:

- What is the distribution of the observed performance by individual peers in a torrent?
- What peer- or group-level properties primarily determine the observed performance by individual peers in a torrent?

The first question reveals how similar (or dissimilar) are the observed performance by individual peers. This in turn determines whether a few peers can properly represent the entire population of participating peers and how they should be selected. The second question explores any potential dominant factor(s) that affect the observed performance by individual peers. To our knowledge, these questions have not been investigated by previous measurement studies on BitTorrent.

In this chapter, we try to answer these two important questions by capturing the observed performance for almost all participating peers in several torrents with different groups of users. We present our methodology to derive peer- and group-level properties of participating peers in a torrent from its tracker log. We also describe various challenges in our approach including the difficulty to accurately estimate the observed performance by all participating peers. To tackle the first question, we examine the distribution of the derived peer- and group-level properties in our candidate torrents and illustrate that both the observed performance and the observed group level properties significantly vary among participating peers. To answer the second question, we investigate the correlation between the observed performance by each peer and both its peer-level and its observed group-level properties using several classic techniques. Our analyses demonstrate that while the performance of each peer has the highest correlation with its outgoing bandwidth, there is no dominant peer- or group-level property that primarily determines the observed performance by the majority of peers. This suggests that the dominant determining factors for the observed performance by individual peers is different. This chapter makes the following contributions: (i) it presents a set of techniques to accurately derive peer-level and group-level properties of all participating peers in a torrent; (ii) it illustrates that the commonly used approach of instrumented clients is inappropriate to characterize peer-level performance in BitTorrent; and (iii) it provides several evidences that the relationship between the peer-level performance and other peer- or group-level properties is non-trivial, and although there are significant correlations, there is no dominant factor that determines peer-level performance.

The rest of this chapter is organized as follows: In Section 5.1., we present a brief overview of BitTorrent to provide the required background for our study. Our measurement methodology, our dataset and our data processing are described in Section 5.2.. We present the peer-level and group-level properties in Section 5.3.. In Section 5.4., we examine the correlation between peer performance and various properties. Finally, Section 5.5. concludes the chapter.

5.1. BitTorrent: An Overview

To provide the proper context for our study, we present a brief overview of those aspects of BitTorrent that are relevant to our measurement and characterization. In BitTorrent, all participating peers that join the system to download the same file are referred to as a “torrent”. All peers in a torrent form a random mesh and incorporate swarming content by pulling their missing segments from connected peers. Peers are generally divided into two groups, seeds and leechers, that have the entire or part of the entire file, respectively. All peers provide content to their neighbors but only leechers need to download content.

BitTorrent features a peer-level incentive mechanism among connected peers called tit-for-tat. This mechanism tends to connect together peers with similar ability to provide content. Therefore, the tit-for-tat mechanism can affect achieved download rate by individual peers. In general, the observed download and upload rates by each peer is limited by its incoming and outgoing access link bandwidth, respectively. However, these rates could be further limited by the user and by cross traffic. Peers can join and leave a torrent in an arbitrary fashion. These dynamics of peer participations (or churn) could also affect observed performance by individual peers.

For each torrent, there is a well known node called *tracker*. The tracker keeps track of all the participating peers in a torrent as well as their download progress. Each peer contacts the tracker when it joins or leaves a torrent, or requires more neighbors. Each peer also periodically (every 30 minutes) reports its total amount of uploaded and downloaded bytes, among other information, to the tracker. The tracker records all these interactions (including peer arrival, departure and periodic updates) in a log file. In the next section, we describe how the tracker log of a torrent can be leveraged to characterize its peer- and group-level properties.

5.2. Measurement Methodology

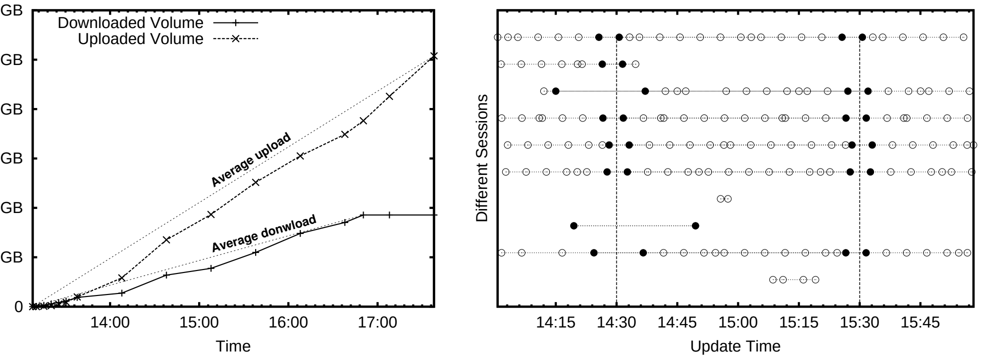
A common approach to study BitTorrent is to run multiple instrumented clients and capture their observed performance. This approach provides detailed information (*e.g.*, access link bandwidth, variations of download rate over short timescales) about observed performance by several peers. However, this approach has two important limitations: *(i)* Since the distribution of observed performance among participating peers is unknown, the observed performance may not provide a representative view of the entire population. *(ii)* This approach does not provide any group-level information (*e.g.*, average content availability, group population) that might have a significant impact on the peer-level performance.

To address this problem, we leverage BitTorrent tracker log to estimate both peer-level properties (namely download and upload rate) for *nearly all* participating peers and key group-level properties (*i.e.*, churn rate, content availability, and group population) that are observed by individual peers. This information allows us to answer our two key questions. It is worth noting that this approach has its own limitations as follows: First, as we explained earlier, each peer sends an update of its download (and upload) progress once every 30 minutes. Therefore, we can only estimate "average" peer-level properties over 30 minute timescale. This implies that variations on download and upload rates over shorter timescale can not be captured by this approach. Second, the tracker log does not contain any information about the connectivity between participating peers (*i.e.*, shape of the overlay topology). Therefore, we are not able to examine the potential effect of content availability among neighbors of a given peer on its performance. Third, the tracker log does not provide any explicit information about the maximum download or upload rate that each peer is able (willing or configured) to achieve. This could affect the accuracy of estimated performance by each peer. We further elaborate on this issue and explain our approach to address this problem in subsection 5.2.3.. In the next two subsections, we describe how peer- and group-level properties are derived from tracker logs.

5.2.1. Deriving Peer-Level Properties

We only focus on two peer-level properties: download and upload rates. Toward this end, we define a *session* as a collection of events associated with a single appearance of a particular peer in a torrent. A complete session starts with a *sign-in* event in the tracker log, continues with several periodic updates, and finally ends with a *sign-out* event. Note that the tracker log is missing "sing-in" (or "sign-out") events of a session if these events occur outside our logging window. A session may also include a *download completion* event which implies that a peer has become a seed. In our study, we only focus on the observed performance by leechers until they complete their download.

The average download (or upload) rate for a particular peer between two consecutive updates is estimated by dividing the increase in the amount of downloaded (or uploaded) bytes during this interval updates by its duration. This leads to several rather short term average upload and download rates (one per update) for each peer. The average download and upload rate during the entire session can be similarly estimated by comparing the first and the last (or the download completion if it occurs during the session) reports. Figure 5.1.a depicts the evolution of downloaded and



(a) Calculating average download and upload rate

(b) Sampling method for group properties

FIGURE 5.1. Capturing peer- and group-level properties

uploaded bytes over time for a single peer. The slope between two consecutive points represents the average download/upload rate for that interval whereas the slope of the line that connects the first and last points represents the average rate across the entire session. This figure clearly shows that: *(i)* This peer completes its download shortly before 5pm but remains in the system as a seed, and *(ii)*its average upload rate is higher than its average download rate.

5.2.2. Deriving Peer-View of Group-Level Properties

Our goal is to derive the average value of key group-level properties, namely population, churn rate, and content availability, that are viewed by individual peers during their session. To achieve this goal, we first derive evolution of these properties over time. We sample the value of these group-level properties at evenly spaced points in time. Figure 5.1.b demonstrates this approach by showing the arrival time of all received updates from each peer (with a circle) on a horizontal line. At each sampling point, we only consider the last report before and the first report after the sampling point for each active session (shown with a filled circle). Given the available content at each peer in these two reports, we can estimate the available content at that peer at the sampling point. Then, we can average the available content across all active peers to estimate average content availability at that sampling point. Counting the number of active sessions at a sampling point provides an estimate for the population of peers at that point of time. Comparing the identity of peers at a sampling point with the last sampling point reveals the number of departed or arrived peers since the last sampling point. Using this information, we can estimate the evolution of these group-level properties during the appearance of individual peers (*i.e.*, a session). Then, the peer-view of these group-level properties can be derived for each peer by averaging their values during that peer's presence as a leecher. More specifically, we focus on the average value of group population, churn rate and content availability during the downloading time of each peer to derive the observed value of these group properties for that particular peer. In essence, peer view of these properties represents the state of the group during the appearance of a peer as a leecher.

5.2.3. Performance Metrics

The main goal of individual peers in a torrent is to maximize their download rate. To determine observed performance by individual peers, we should measure their ability to utilize their access link bandwidth, *i.e.*, the ratio of average download rate to the maximum rate that a peer is able and willing to receive content (*i.e.*, its physical, available or configured incoming bandwidth). However, the tracker log does not provide any explicit information about incoming access link bandwidth of individual peers. Therefore, we use the maximum value of per-interval download rate for each peer as an estimate for its access link bandwidth. Since the measured download rates are averaged over 30 minute intervals, they provide a "loose" lower bound for maximum download rate. Using this estimate of incoming access link bandwidth, we define the following two performance metrics for each peer:

- *Access Link Utilization*: The ratio of average download rate to its maximum value during a session estimates the utilization (or relative performance) of a peer.
- *Variability of Download Rate*: The ratio of standard deviation of per-interval download rate to its average value during a session represents the normalized variations or the variability of observed performance by each peer.

The first metric accurately captures performance of each peer but it is sensitive to the estimated access link bandwidth for individual peers. The second metric does not depend on access link bandwidth but it is a rather indirect measure of performance.

5.2.4. Data Set

We have examined the tracker logs for more than 4185 torrents and selected three torrents from different user communities, namely RedHat (RH) and Debian (DE) Linux distributions, and a 3D Game software(GA). Table 5.1. summarizes the characteristics of these three torrents. This table shows that the tracker logs have been collected at different points of time that are at least a few weeks long, and have different population and different number of sessions. The diversity across these tracker logs enables us to determine whether our findings are rather common or specific to a particular torrent. We have conducted several sanity checks on tracker logs to identify any potential error in our dataset. For example we examined whether reported amount of download/upload data by all peers always monotonically grows. We discovered that a small fraction of (potentially buggy) clients do not pass this

TABLE 5.1. Characteristics of three selected torrents

Community	#Start Time	End Time	#Sessions	Max. Pop.
Red Hat	3/2003	8/2003	170814	3684
Debian	2/2005	3/2005	139736	91
3D Games	10/2004	12/2004	195660	1530

condition. We also noticed that there are some gaps in some tracker logs (*i.e.*, no event is recorded for a couple of hours). This could occur when tracker becomes unreachable for any reason. We have removed information about any misbehaving session from our logs and only focus on the portion of logs that does not contain any gap to avoid any significant error in our analysis. We also remove all the short-lived sessions (with uptime less than 30 minutes) since their performance could be significantly affected by their short stay in the system.

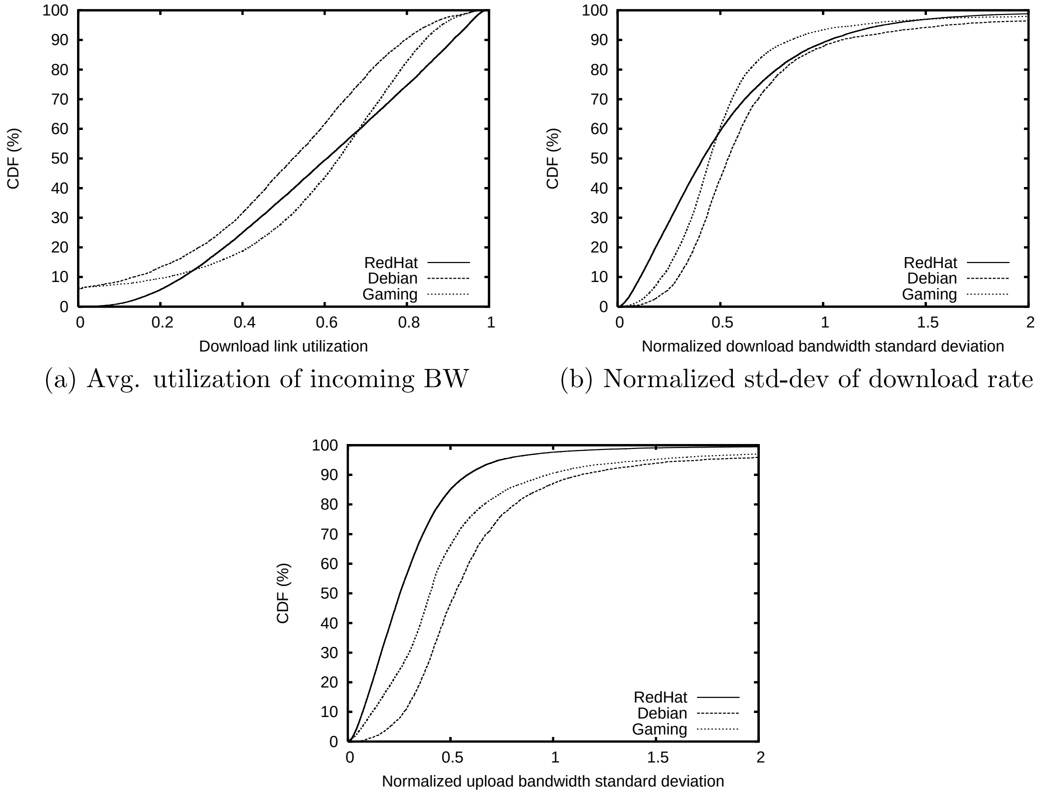
5.3. Distribution of Observed Properties

In this section, we examine the stability of peer-level and group-level properties in our three candidate torrents. Toward this end, we try to answer the first question that we raised earlier as follows: *What is the distribution of the observed peer-level and group-level properties among participating peers in a torrent?* Note that participating peers in a torrent may appear at different points of time during our long measurement period. We explore this issue for peer-level and group-level properties in the following subsections.

5.3.1. Peer-Level Properties

Figures 5.2.a and 5.2.b present the distribution (CDF) of two peer-level performance metrics, the average utilization of incoming access link and the normalized standard deviation of download rate, among participating peers in all three torrents where each torrent is labeled with its corresponding community. These distributions reveal several interesting points as follows: First, despite the differences among these torrents, the distribution of each performance metric has an interestingly similar shape for all three torrents. Second, around 10% of participating peers in Figure 5.2.b exhibit significant variations in their download rates, *i.e.*, experience poor performance. If we exclude these low-performing outliers from Figure 5.2.b, both figures depict a pretty smooth distribution without any dominant mode. This implies that the probability of experiencing a certain level of performance (between 0 and 1) is rather similar. *In a nutshell, our results from all three torrents illustrate that participating peers in a torrent experience a rather diverse performance with a roughly uniform distribution.*

To explore the behavior of other peer-level properties, Figure 5.2.c shows the distribution of normalized standard deviation of upload rate for all three torrents. These distributions are very similar to those for normalized download rate (in Figure 5.2.b) which suggest that the contribution of participating peers into their torrent is rather diverse. Furthermore, the contribution of participating peers in the RedHat torrent is higher than the Debian torrent, and in the Debian torrent is higher than the Gaming torrent. To explain this we note that participating users in the RedHat and Debian torrents are usually tech-savvy clients that have nodes with higher bandwidth connectivity and processing capabilities. The similarity between the distribution of normalized download and upload rates suggest that they might be correlated. We will further examine this issue in the next section.



(a) Avg. utilization of incoming BW

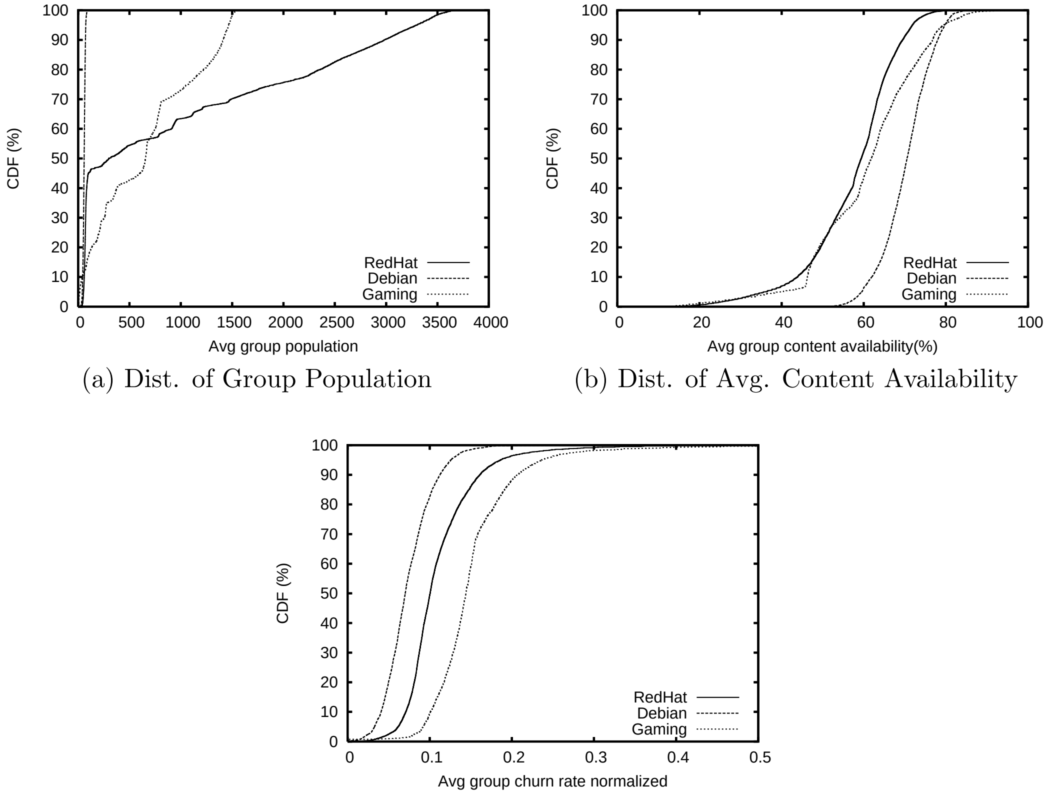
(b) Normalized std-dev of download rate

(c) Normalized standard deviation of upload rate

FIGURE 5.2. Distribution of performance metrics across all peers for three torrents

5.3.2. Peer-View of Group-Level Properties

We now turn our attention to the observed group-level properties and examine their variability among participating peers in a torrent. Figure 5.3.a, 5.3.b, and 5.3.c present the distribution of peer-view of three key group-level properties among participating peers in our three candidate torrents. The distribution of average group population (in 5.3.a) is clearly different across three torrents. The RedHat torrent contains the initial flash crowd where the population of peers varies between 200 to 3500 peers, and around 55% of peers complete their download during this initial phase. The Debian and Gaming tracker logs do not contain the flash crowd phase.



(a) Dist. of Group Population

(b) Dist. of Avg. Content Availability

(c) Dist. of Avg Churn Rate

FIGURE 5.3. Distribution of peer-view of group-level properties for three torrents

The observed group population by peers in the Debian torrent changes between 50 to 1500 whereas the population of the Gaming torrent remains rather stable around 50 peers. In short, the observed average group population among participating peers in these three torrents exhibit significantly different characteristics. Despite this difference in group population, the distribution of observed content availability and churn among participating peers in each torrent is rather similar (*i.e.*, distribution is not too skewed). More specifically, peers in each torrent have experienced around 50-70% average content availability among their coexisting peers in the system (not necessarily among their neighbors), and the average observed churn rate by peers is different across these torrents but is rather similar among peers in each torrent. In summary, our results show that the participating peers in a torrent do not experience a similar performance. Except for group population, other peer- and group-level properties exhibit similar overall trends across different torrents.

5.4. Identifying Underlying Factors

In this section, we tackle the second question that we raised earlier as follows: *What are the peer- or group-level properties that primarily determine the observed performance by individual peers in a torrent?*. To answer this question, we derive the observed performance by each peer (based on both performance metrics) along with its peer- and group-level properties. Given this information for all peers in a torrent, we leverage several classic techniques to identify (either qualitatively or quantitatively) any correlation between each performance metric and the following key properties that we discussed in the previous section: upload rate, group size, content availability and churn. Simple techniques such as scatter-plots did not reveal any clearly visible correlation between the observed performance and peer- or group-level properties. Therefore, we focus on more elaborate techniques in this section.

5.4.1. Linear Regression

We perform linear regression (using Splus) as a classic statistical technique to establish a linear relationship between observed performance by each peer and its main peer- and group-level properties(*i.e.*, median upload rate, average group population, average content availability and average churn rate). The derived model also quantifies the impact of each property on the overall performance. To minimize the effect of outliers, we remove any session whose properties are within the top or bottom 10% of observed range of values.

TABLE 5.2. Linear regression results for RedHat torrent. (Coefficient, P-Value)

Model	R-square	outbw.50p	avg.grp.pop	avg.grp.cont.avail	avg.grp.churn
util	0.0651	0.0091, 0	-0.1206, 0	0.3493, 0	0.0015, 0
util-log	0.0603	0.0965, 0	-0.0311, 0	0.4367, 0	0, 0
util-step	0.0603	0.0965, 0	-0.0309, 0	0.4358, 0	removed
sdev	0.0709	-0.0142, 0	0.2245, 0	-0.3344, 0	-0.0029, 0
sdev-log	0.0741	-0.1585, 0	0.0778, 0	-0.6486, 0	-0.0005, 0.0095
sdev-step	0.0741	-0.1585, 0	0.0778, 0	-0.6486, 0	-0.0005, 0.0095

Each row of Table 5.2. presents the coefficients for different properties that represent the derived linear model by this technique for RedHat torrent. The results for other torrents are similar. For each torrent, we examined each performance metric in the following three scenarios in a progressive fashion: *(i)* The base model that relates a performance metric with all properties, *(ii)* the model that relates a performance metric to the log value of some properties (population and upload rate), and *(iii)* same as step *(ii)* but we use the "step" function in Splus to simplify the model by removing least important factors when possible. The goal in examining log value of properties is to reduce the range of values for properties which in turn could reveal any non-linear relationship that might exist as well. Each row also includes "R-Squared" value which estimates the percentage of sessions that can be properly predicted by the derived model.

As Table 5.2. indicates, all R-square values are smaller than 0.1 (*i.e.*, models can predict performance in less than 10% of sessions). In essence, this table provides a clear evidence that there is no simple linear (or non-linear) model that properly captures the relationship between the observed performance and other examined properties for individual peers. Therefore, instead of deriving a model that incorporates all the properties, we explore the pairwise correlation between the observed performance and each property which is easier to observe in the next subsection.

5.4.2. Spearman's Rank Correlation

We use Spearman's rank correlation test as powerful technique to quantify the degree of correlation between each performance metric and peer- and group-level properties. Spearman's rank correlation coefficient is a non-parametric measure of correlation that assesses how well an arbitrary monotonic function could describe the relationship between two variables, without making any assumptions about the frequency distribution of the variables. Table 5.3. presents the Spearman's rank correlation coefficient between our two performance metrics and the following properties for all three candidate torrents in Table 5.1.: normalized standard deviation of upload rate (dev.upload), average group population (pop), average content availability (cont), and average churn rate (churn).

This table illustrates several interesting points: First, despite difference between three torrents, both performance metrics appears to have the highest correlation with the upload rate. This suggests that the variability of upload rate has the highest effect on the observed performance which in turn implies that tit-for-tat mechanism has the most noticeable impact on performance. Furthermore, in all three torrents, the correlation coefficient between our two performance metrics and outgoing bandwidth have a close absolute value with opposite signs. Second, aside from the upload rate, the effect of other parameters on observed performance by individual peers seems to vary across different torrents and in some cases between different performance metrics. In the RedHat torrent, both performance metrics have a relatively stronger correlation with group population and churn. This could be due to the fact that the

TABLE 5.3. Spearman's rank correlation coefficient (3 torrents)

Torrent	**Perf.**	dev.upload	Pop	Cont	Churn
RH	inbw.util	-0.46	-0.13	0.05	-0.12
RH	inbw.nsdev	0.49	0.20	-0.03	0.19
DE	inbw.util	-0.42	-0.02	0.10	-0.02
DE	inbw.nsdev	0.47	0.03	-0.10	0.00
GA	inbw.util	-0.36	-0.05	0.04	-0.05
GA	inbw.nsdev	0.47	0.14	-0.11	0.14

log for this torrent contains the initial flash crowd phase where more than half of the captured sessions lie. Average content availability has a relatively larger coefficient for both metrics in the Debian torrent. This is most likely due to the small population in this torrent. Finally, in the Gaming torrent, the first performance metric (access link utilization) has a small and comparable coefficient for all three properties while the second performance metric (normalized standard deviation of download rate) has larger coefficients for all three properties. Clearly, the potential error in estimating the incoming access link bandwidth could affect the derived coefficients in our analysis. However, since the coefficients for upload rates and both metrics are similar, we believe that the impact of error on the largest coefficients is rather small. In summary, our results suggest that upload rate by individual peers (*i.e.*, its contributed upload rate to the system) has the primary effect on their observed performance. This finding is based on both performance metrics and across all three torrents.

5.5. Summary

In this chapter, we examined a repeated claim that BitTorrent can provide high performance (*i.e.*, download rate) to participating peers in a torrent. We derived peer-level performance along with observed peer- and group-level properties among all peers in three different torrents. We showed that the distribution of performance among participating peers in the a torrent is roughly uniform. We also investigated the impact of various properties on the observed performance by individual peers. First, we try to build a linear model for the observed performance using linear regression based on various peer- and group-level properties. The group-level properties are essentially specifications of the particular torrent during the life time of a particular peer. We include torrent population, average content availability, and average churn. The resulting R-square value in all cases is below 0.1 which means that the linear regression cannot model the observed performance with reasonable accuracy. Next we use Spearman's rank correlation in order to find any rank correlation among group- and peer-level properties with the observed performance. The only parameter showing significant correlation with the performance metrics was the average upload rate of the peer.

Main findings are the following: *(i)* There is no clear relationship between peer-level performance and main peer- and group-level properties, *i.e.*, the relationship could significantly vary among peers, and *(ii)* average upload rate (*i.e.*, contribution) of individual peers has the highest correlation with its observed performance. This suggests that the tit-for-tat mechanism in BitTorrent is the primary factor that affects peer-level performance. These findings reveal that a common approach of using a few instrumented clients does not provide a representative view of BitTorrent behavior. Instead, a more global view must be considered in order to derive a reliable and general conclusion.

During the past chapters, we presented three measurement studies on P2P systems. First, we studied the Gnutella P2P application and studied its evolution over time. Next we introduced a sampling technique, namely, RDS, and showed how it can be used to efficiently capture peer properties of a large scale P2P system. In this chapter we focused on the performance measurement in a P2P application, namely, BitTorrent and tried to establish relationships between the observed performance and the properties of the peer itself, as well as properties of the group during the peer's lifetime.

After visiting the *P2P overlay* and the involved problems, we will turn our attention to the *Internet underlay.* In the next chapter, we focus on the Internet infrastructure and its building blocks, namely, Autonomous Systems (ASs). We use our P2P measurement techniques and data to capture the geographical properties of the ASs and their connectivity.

CHAPTER VI

AS-LEVEL UNDERLAY: GEOGRAPHICAL MAPPING

Most of the content from this chapter has been adopted from my previously published paper [104] co-authored with Dr. Nazanin Magharei, Prof. Reza Rejaie, and Dr. Walter Willinger. The experimental work is mine with some assistance from Dr. Magharei and the text has been contributed by myself and the co-authors. I am grateful to Prof. Matthew Roughan for his invaluable ideas contributing to this project. Kad and BitTorrent datasets are kindly provided by Ghulam Memon and Dr. Ruben Cuevas. I acknowledge Kaveh Kazemi for extracting PoP information for a large number of ASs from the web. I also thank Maxmind and Hexasoft for providing their IP geo-location databases.

As a network of networks, the Internet consists of some 30,000 inter-connected Autonomous Systems (ASs). This AS-level topology has been the focus of much research in the past decade, with studies that range from measurements and inference [85] to modeling and analysis [83] and the development of synthetic topology generators [84]. In fact, much of the research in this area has been fueled by large-scale data collection projects (*e.g.*, [64, 93, 120]) that have resulted in a high volume of readily available BGP-based or traceroute-based measurements. These datasets have been used to infer the Internet's AS-level topology as a graph where nodes are ASs and edges indicate business relationships (*e.g.*, customer-provider, peer-to-peer) between ASs.

More recently, this graph view of the AS-level Internet has been questioned. First, there has been an increasing awareness that the available BGP- or traceroute-based measurements are of limited quality to obtain an accurate and complete picture of the AS-level connectivity structure of today's Internet (*e.g.*, see [10, 95] and references therein). Second, the models that this graph view has motivated are largely descriptive in nature and essentially agnostic to the main forces responsible for shaping the structure and causing the evolution of this inherently virtual rather than physical topology of the Internet.

Partly in response to this criticism, alternative approaches to study the AS-level Internet have been advocated that align more closely with the real-world business relationships and practices encountered in the logical fabric of the Internet (see for example [26] and [39] and references therein). These recent efforts often start with the realization that ASs are not generic nodes but are entire networks that operate for a purpose and have a rich internal structure. Depending on an AS's size, its network interconnects a number of geographically dispersed points-of-presence (PoPs), where it connects to its customers or interconnects with other networks, either directly or via Internet eXchange Points (IXPs). The importance of AS geography (*i.e.*, geographic coverage or reach, number and location of PoPs, presence at IXPS) is further highlighted by the fact that the peering contracts of many ASs list explicit and geography-specific requirements for potential peering partners. For example, AS X will only peer with AS Y if Y's geographic reach is sufficiently large, or X and Y have a certain number of overlapping PoP locations, or X and Y are both present at a certain number of IXPs. Unfortunately, little is known in general about the geography of most ASs, with the possible exception of their presence at IXPs that was specifically examined [10].

In this chapter, we outline a promising approach to tackle the problem of AS geography; that is, inferring an AS's geographic coverage (*geo-footprint*) and identifying its likely PoP locations. Our approach is complementary to the traditional BGP- or traceroute-based method of inferring AS-level connectivity, in both perspective and type of data used. First the traditional approach is known to perform in general increasingly worse the closer to the "edge" of the network (*i.e.*, end users) the measurements are made [27]. However, our approach starts at the "edge" (*i.e.*, "eyeballs") and experiences increasingly more difficulties as we move away from the edge towards the core of the Internet. In terms of data, instead of using BGP or traceroute data, our approach relies on the geographical location of end users or "eyeballs" (*i.e.*,, IP addresses) that are associated with an AS. In particular, the starting point of our work is a dataset consisting of the IP addresses of about 48 million users of three popular P2P applications that map to a total of 1233 "eyeball" ASs. Our main contributions in this chapter are the following:

- Considering end-users as pinpoints, we propose a general methodology for mapping the geographical footprints of eyeball ASs. We map each end-user's location using their IP addresses and then build a user density function for each AS using KDE mathematical tool (Sections 6.1., 6.2.).
- We propose a method to identify the likely PoP locations of an eyeball AS by associating the local maxima of the user density function with close-by cities within the geographical footprint of the AS (Section 6.3.) and validate our approach using published PoP locations of numerous eyeball ASs.
- We perform a case study using our inferred PoP locations and the state of the art AS connectivity information in order to explore peering relationships between eyeball ASs. The observation shows that the peering relationship at the "edge" of the network are very complex and simple eyeball ASs actively

peer at local and remote Internet Exchange Points while maintaining a rich upstream connectivity.

The question of how to leverage the geo-properties of an eyeball AS to predict likely scenarios of how the AS connects to the rest of the Internet is left for future work. In view of our preliminary findings, a major challenge will be to explain the observed rich connectivity structure of eyeball ASs and characterize it in a quantitative manner.

6.1. Our Approach: An Overview

The basic idea of our approach is to use the location of end-users (*i.e.*, customers) of an AS to infer the AS's geographical reach (geo-footprint) as well as its PoP locations. To achieve this goal, our method consists of the following four steps:

- *Sampling end-users*: We collect a large number of IP addresses associated with Internet users.
- *Mapping end-users to locations*: We map individual IP addresses (or users) to their geo-location.
- *Grouping end-users by AS*: We use BGP information to group users to their corresponding ASs.
- *Estimating AS geo-footprints*: We leverage the collection of geo-locations of end-users associated with an eyeball AS to determine the geo- and PoP-level footprint of that AS.

There are three main reasons for our focus on eyeball ASs. First, the geo-features and connectivity of eyeball ASs indicate how end-users connect to the rest of the Internet. These eyeball ASs are not adequately visible to traceroute- or BGP-based approaches. Second, the accuracy of IP-geo mapping tools is significantly higher for IP addresses associated with end-users compared to infrastructure nodes [123]. Lastly, it is feasible to obtain a collection of IP addresses associated with end-users from eyeball ASs. Next, we provide further details for the first three steps of our approach. The last step, estimating geo- and PoP-level footprints, is described in Sections 6.2. and 6.3..

Sampling End-users: We crawl three large-scale P2P applications (*i.e.*, Kad, BitTorrent and Gnutella) during the months of January to June of 2009 to obtain more than 89.1 million unique IP addresses associated with end-users (peers) of these applications.

Mapping Users to Locations: To estimate the geo-location of each IP address, we examined several IP geo-location tools and databases and selected *GeoIP City from Maxmind* [86] and *IP2Location DB-15 from Hexasoft* [63] because of their reputation and coverage. Each of these databases map any IP address to a geo-location record with the following format *(city, state, country, longitude, latitude)*. The resolution of the provided coordinates is zip codes in each city, *i.e.*, all users in a given zip code are mapped to the same coordinates. We eliminated roughly 2.4M peers for which at least one of the databases did not provide city-level location. Since the two IP-geo mapping databases are from independent sources, we use the difference between their reported locations for each peer as a measure of error in IP-geo mapping [1]. We use GeoIP City as the main reference for IP-geo mapping in our analysis and use IP2Location as a second reference to estimate the error in IP-geo mapping. Using this notion of error, we remove all IP addresses whose error is larger than the diameter

[1] While this measure of error may not be accurate, it provides a first-order approximation of geo error and could be useful to conservatively remove problematic peers with potentially large error in their geo-location.

TABLE 6.1. Target eyeball ASs profile.

	#Peers by source(k)			#ASs by level		
Region	Kad	Gnu	BT	City	State	Country
NA	1218	8984	1761	36	162	129
EU	18004	2519	2529	60	76	292
AS	17865	1606	1016	117	35	134

of typical metropolitan area, around 100km. We further elaborate on the selection of this threshold in Section 6.2..

Grouping Users by AS: We also group the users based on their AS affiliation using archived BGP tables from the routeviews[93] database collected during the same time period that our P2P data was gathered. To ensure a minimum density of samples in each AS, we eliminate all ASs with less than 1000 peers.

Target Dataset: Conditioning our dataset based on error in geo-location and density of sampled peers per AS significantly decreases any noise that could affect our analysis. However, it also reduces the total number of peers to 48 million and the corresponding number of eligible eyeball ASs to 1233. We call this set of ASs our *target dataset*. Given the location of all peers associated with an AS, we can broadly classify all ASs in this target dataset into city-, state-, country-, continent-level, or global ASs by identifying the smallest geographical region that contains a large majority (>95%) of the associated peers. Table 6.1. summarizes the number and level of our target ASs in North America(NA), Europe (EU) and Asia (AS).

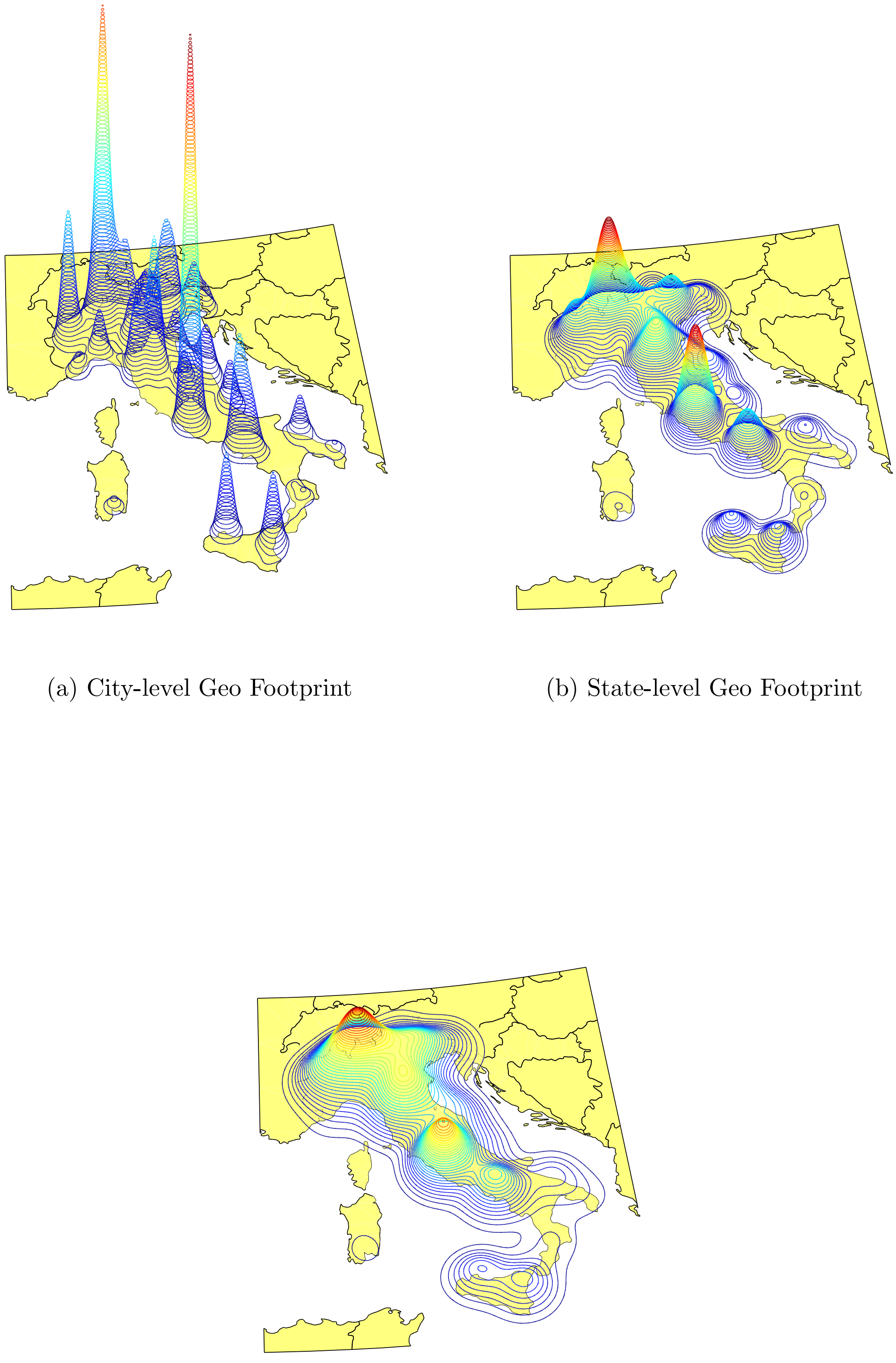
6.2. Estimating Geo-Footprint

Given the locations of peers associated with an eyeball AS, our first goal is to infer the geographical region(s) where the AS offers service to end-users (*i.e.*, its geo-footprint) and estimate the density of users throughout the identified regions. We use a Kernel Density Estimation (KDE) [18] method with a Gaussian kernel function to estimate the probability density of customer population for an eyeball AS based on the locations of peers associated with that AS. More precisely, we place a (bivariate) kernel function with a predefined bandwidth at the geo-location of individual users of the AS. The aggregation of these kernel functions forms a function that estimates the overall user density over the map for each AS as shown in Figure 6.1.. The largest contour of the aggregate density represents the geo-footprint of the AS at certain levels of resolution and may consist of one or multiple partitions. The geo-footprint of an AS clearly highlights the area within a state, country, or continent where an AS offers service, and its pronounced peaks indicate the main places with high user concentration throughout the covered region. This geo-footprint provides useful information about the services offered (*e.g.*, residential vs. retail) and connectivity provided (many vs. a few peaks) by individual eyeball ASs.

The KDE method presents a weighted average across close-by peers that serves two purposes. First, averaging smooths out the effect of error in IP-geo mapping across close-by users and provides a more reliable estimation of user density. Second, averaging offers a more aggregate (lower resolution) view (city- or state-level) of the users that is typically more useful than a detailed user-level view for assessing the geo-footprint of an AS.

6.2.1. Setting the Kernel Bandwidth

The level of smoothing (*i.e.*, scope of averaging) performed by the KDE method is directly controlled by the bandwidth of the kernel function. Increasing the bandwidth leads to aggregation over a larger geographical region that has two important effects. First, it results in a coarser resolution and thus less accurate estimation of the geo-



(a) City-level Geo Footprint

(b) State-level Geo Footprint

(c) Country-level Geo Footprint

FIGURE 6.1. 3D visualization of user density from KDE method for AS 3269 (in Italy) using 2.2M samples with kernel bandwidth of 20km, 40km, and 60km.

footprint for an AS. In fact, the bandwidth of the kernel function can be viewed as a tuning parameter that offers a multi-resolution view of an eyeball AS's geo-footprint. Figure 6.1. clearly demonstrates how increasing bandwidth can change the resolution of the geo-footprint from city- to region- and finally country-level. Second, averaging smooths out the variations in user density which makes the distinction of (smaller) peaks more difficult. It is therefore desirable to set the bandwidth so that the following two conditions are satisfied: *(i)* the resulting geo-footprint should have the desired resolution, and *(ii)* the expected geo-location error across the provided users should be filtered out. In summary, the larger value of the minimum bandwidth required by each one of these conditions determines the proper bandwidth value for the kernel function. For example, samples with a large *geographical mapping error (geo error)* cannot provide reliable city-level resolution of an AS's geo-footprint.

In our analysis, we focus on the *city-level* resolution because it provides the most useful view for detecting the main concentration point of users in order to infer likely PoP locations for each eyeball AS. To achieve this goal, the bandwidth should be larger than the average radius of a city which is around 30-35km. We set the bandwidth of the kernel function to 40km to achieve aggregation over a slightly larger region and avoid multiple peaks over a single city (*e.g.*, a separate peak for each zip code).

To determine a lower bound for the bandwidth based on geo error, we could set the bandwidth for each AS to the 90th percentile of geo error across all peers in that AS. This would result in an AS-dependent bandwidth selection. Instead, we remove all the ASs whose 90th percentile of geo error is larger than 80km. This is the main justification for removing all peers with geo error larger than 80km from our initial dataset. This strategy allows us to set the bandwidth to 40km for all ASs to obtain a city-level resolution. This choice simplifies the comparison of geo-footprints across different eyeball ASs

6.3. Estimating PoP-Level Footprint

A geo-footprint of an eyeball AS can be summarized or represented by the list of major cities where significant portions of its customers are located. Intuitively, each AS must have a proportional level of presence (*e.g.*, PoP) in areas where there is a high concentration of customers. Therefore, this representation of an AS geo-footprint offers a reasonably reliable view of its PoP-level infrastructure that we call *PoP-level footprint*. Since eyeball ASs usually connect to their provider, peering and customer ASs at their PoPs, the PoP-level footprint also reveals valuable information about the location(s) of connections between related ASs.

6.3.1. Estimating PoP Coordinates

To extract the PoP-level footprint of an eyeball AS from its geo-footprint, we proceed as follows. First, we identify the geo-coordinates of all the local maxima $D(i)$ (*i.e.*, peaks) in the estimated density function and determine the density value of the highest peak (D_{max}). Next we select all the peaks $D(i)$ with a relatively large density compared to the highest peak, *i.e.*, ($D(i) > \alpha^* D_{max}$), where α is a threshold that determines the range of density values that are considered for PoP identification. We set α to 0.01 to conservatively select peaks with a density of at least two orders of magnitude below D_{max}.

6.3.2. Mapping PoPs to Cities

The coordinates of identified major peaks may not directly map to a city due to the combined effects of selected bandwidth, threshold for peak selection (*i.e.*, α), and the distribution of user population around each city. To address this issue, we map identified peaks to a particular city in a "loose" fashion as follows: we assume that PoPs are more likely to be located in the most populated city of a given region. For each identified peak, we examine a circular region with a radius equal to the selected kernel bandwidth around the location of the peak and map the peak to the city with the largest population in that circular region. Otherwise, we report "no city" for a peak. Using small values for α may result in an error whereby minor peaks of user density get selected at locations where a small number of users are randomly clustered due to their geo error.Using a proper α threshold, we can filter out such peaks if the selected location is not in the required vicinity of any city with sufficiently large population. The resulting PoP-level footprint obtained by this process consists of a list of cities sorted by their associated user density where PoPs of an eyeball AS are likely to be located. The user density of each PoP quantifies the level of presence of an AS in that city. For example, PoP-level footprint of an AS number 3269 that serves Italy is as follows: [Milan (.112), Rome (.083), Naples (.076), Florence (.052), Venice (.048), Turin (.045), Catania (.031), Palermo (.028), Bari (.024), Ancona (.023)].

6.3.3. Bias in Sampling Different Locations

We crawl peers in major P2P applications to sample customers of eyeball ASs from different geo-locations. Uneven penetration of P2P applications among Internet users in different ASs and locations could introduce bias to our samples. However, it is generally difficult to clearly distinguish the small market share of an AS from low penetration of a P2P application in a particular city. This potential bias can be qualitatively considered at two different levels.

1) Mild Bias: This scenario occurs when the fraction of sampled peers for an AS A in city C has a noticeable density ($D_A(C) > \alpha * Max(D_A)$) but is disproportional with respect to the total number of AS customers in C. In this case, the derived PoP-level footprint of the AS includes city C as a PoP but the density value associated with C is inaccurate.

2) Significant Bias: A significant bias in collected samples could result from having a negligible (or zero) fraction of samples from a particular PoP location for a given eyeball AS. In this case, our approach does not discover that PoP location. However, for an AS with a sufficiently large number of samples, the probability of not capturing a major PoP (with a large number of customers) should be rather small. We do not examine sampling bias in this study and leave this for future work.

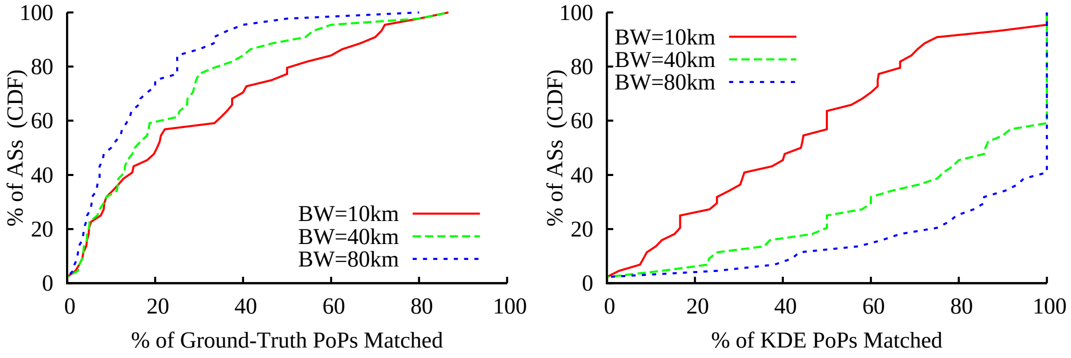
Another issue is whether the strategy for IP-geo mapping leverages the location of PoPs for each AS to estimate the location of end-users in that AS? In this case, our approach simply identifies the PoP locations that were used for IP-geo mapping. Our private communication with maxmind.com confirmed that the IP-geo mapping strategy relies on the information provided by users through online surveys, and information from Internet registries and ISPs. Since the actual location of PoPs for each ISP is unknown and thus not considered for determining the geo location of users in an ISP, the identified PoP locations by our approach are not affected by the mapping strategy used for each dataset.

6.4. Evaluation

This section summarizes the preliminary evaluation of our proposed technique. Towards this end, we collect the reported PoP information of some eyeball ASs on the Web as the ground truth for validation. Unfortunately, collecting this information is a rather tedious task since many ISPs do not post this information online or do not use a consistent terminology or approach for listing these PoPs. For example, some ISPs may consider their access points as a PoP or list their PoPs of their peering ISPs as their own.

We focused on 672 state- or country-level ASs in our target dataset and searched the Web for their PoP information. We were able to identify PoP information for a total of 45 eyeball ASs (10 state-, 33 country-, and 2 continent-level) across North America and Europe [2]. We consider this information as ground truth and call this our reference dataset. Overall, our approach on average identified 31.9, 13.6 and 7.3 PoPs per AS with kernel bandwidth of 80km, 40km and 10km, respectively. The average number of reported PoPs per AS in our reference dataset is 43.7. We match a discovered PoP location by our technique for each AS with a reported PoP locations in the reference dataset if their relative distance is less than the radius of a city (*i.e.*, 40 km), *i.e.*, matching PoPs at the city level. Figure 6.2.a depicts the distribution of the percentage of PoPs in the reference dataset that are matched with the identified PoPs by our techniques using different bandwidth values. When kernel bandwidth is 40km, for the bottom 60% of ASs, the fraction of matched PoP locations in the reference dataset is less than 20%. However, this ratio is larger than 50% for the top 10% of ASs. Furthermore, this figure suggests that using lower bandwidth generally results in mapping a larger number of PoPs in the reference dataset.

[2] This information will be posted online at http://mirage.cs.uoregon.edu/AS2PoP



(a) Percentage of ground-truth PoPs found by our method

(b) Percentage of PoPs reported by our method that match a ground-truth PoP

FIGURE 6.2. Validation of our technique with the reported PoP information online

Figure 6.2.b illustrates the opposite view by showing the distribution of the percentage of discovered PoP locations by our technique for each AS that match a reported PoP in the reference dataset. This figure reveals that with the bandwidth of 80km, 60% of ASs exhibit perfect match. Interestingly, decreasing the value of kernel bandwidth to 40km and 10km rapidly drops the percentage of perfect match to 41% and 5%, respectively. *Collectively, these results indicate that using larger kernel bandwidth leads to a smaller but more reliable set of PoP locations for most ASs.*

Our preliminary examination revealed that the following factors appear to cause the mismatched PoPs: First, some eyeball ASs seem to use certain PoPs in locations away from their regular customers to connect to provider (or peering) ASs. Since these ASs do not serve end-users, our approach is not able to identify them. Second, some eyeball ASs have a few PoPs within a relatively short distance. Using the KDE approach especially with moderate to large bandwidth does not distinguish these PoPs. As part of our future work, we plan to use different kernel bandwidth and determine these PoPs based on the relative distance and user density of associated peaks with different bandwidths. Third, we might have mis-interpreted a non-IP PoP as valid PoP from the obtained information online or a PoP location might be missing due to the obsolete online information. We plan to explore these issues in our future work

We have also compared our discovered PoP locations with the PoP coordinates reported in a recent traceroute-based study by the DIMES project [121]. The overlap between the two datasets consists of 226 eyeball ASs across EU and NA. While for those common eyeball ASs, our approach identified 7.14 PoPs per AS on average (with bandwidth=40km), DIMES reports only 1.54 PoPs per AS. We match a discovered PoP location by our technique for an AS with a reported PoP coordinates in the DIMES dataset within 40km distance. Our results show that for 80% of eyeball ASs our identified PoPs are a clear superset of reported PoPs by the DIMES project.

6.5. AS Connectivity at the "Edge": A Case Study

Having derived the geo-footprint and PoP locations of eyeball ASs, we next examine what this information may enable us to say about how these eyeball ASs connect to the rest of the Internet. Our comparisons are made against the current state-of-the-art "best effort" ground truth for AS-level Internet connectivity and is provided by two different datasets. For customer-provider relationships, we rely on either the CAIDA AS data from the Ark project [64] or the UCLA data from the Cyclops project for peer-to-peer relationships at IXPs, we consult the dataset produced by the IXP mapping work described in [10].

To illustrate the challenges of making any claims about real-world AS-level connectivity at the "edge" of the network, we present a case study involving a metropolitan-area eyeball AS in Europe. Specifically, we consider AS8234 (RAI -

Radiotelevisione Italiana). Based on our data, this AS has 3,000 P2P users whose geo-locations are all mapped to the city of Rome. As a city-level eyeball AS, we expect it to have one or two regional or country-wide upstream providers. Examining the geo-footprints of some of our Italy-wide eyeball ASs, a natural choice of such a provider is AS1267 (Infostrada) for which we observe 1470K P2P users and obtain PoP locations across Italy, including Rome. The large number of P2P users for this ISP suggests that its major business is selling Internet connectivity to residential customers across Italy, and a look at the company's website confirms this. Expecting at least one alternative connection of RAI to the rest of the Internet, plausible options include another upstream provider (possibly with more global reach than Infostrada) or peering at the Rome IXP NaMEX with a selected number of tier-2 ISPs.

However, when comparing against the best effort ground truth which we validated by performing a set of selective traceroute experiments, we encounter a substantially more complex AS-level connectivity picture for RAI. For one, this Rome-based eyeball AS has a total of five upstream providers: Infostrada (as expected) and Fastweb, two Italy-wide ISPs; Easynet and Colt, two service providers with global reach; and BT-Italia, Italy's legacy ISP. Moreover, while RAI is not present at the Rome IXP, it is a member of the Milan IXP MIX and peers there with three other ASs (*i.e.*, GARR - the Italian academic and research network, ASDASD - an Italian network provider, and ITGate - an Italian Internet service company). The two unexpected findings are the richness of upstream connectivity of this eyeball AS and its decision to peer remotely at MIX rather than locally at NaMEX.

Thus, when trying to determine the actual upstream connectivity of eyeball ASs such as RAI, one quickly run into a bewildering web of real-world peering relationships [92, 46, 81, 80]. In some cases, a partial explanation of this richness in

AS connectivity may be the separate treatment of residential and business customer traffic; *e.g.*, residential traffic is carried by one upstream provider, while commercial traffic is sent on to a different provider. In the case of RAI, having the legacy ISP BT-Italia as an additional provider may be more a historical artifact than a strategic business decision. Dual connectivity to upstream providers with global reach may again be a strategic decision based on the eyeball's business model. With respect to RAI's remote peering at MIX, it is worth pointing out that while one of its peering partners there (*i.e.*, GARR) is also present at the Rome IXP, the two other networks (*i.e.*, ASDASD and ITGate) are not members of NaMEX. This suggests that the ability for RAI to peer with the latter two networks is important enough to forgo a cheaper local solution over a more expensive remote peering arrangement.

This example of a simple eyeball AS illustrates the challenges associated with trying to leverage the geo-properties of eyeball ASs to predict and ultimately explain their connectivity to the rest of the Internet.

6.6. Summary

In this chapter, we targeted the problem of AS geography, *i.e.*, inferring the geographic coverage of an AS and identifying its likely PoP locations. The main contributions of this study can be summarized as follows: *(i)* using our captured snapshots from popular P2P applications including millions of user IP addresses, we propose a general methodology for mapping geographical footprints of eyeball ASs; *(ii)* using the locations of all P2P users associated with an AS, we form the user density function for each eyeball AS, through the use of KDE mathematical tool; *(iii)* we present our hypothesis that the local peaks of the resulting user density function for each AS, are correlated with the PoP locations of that AS; *(iv)* we successfully validate our hypothesis using published PoP locations for a set of eyeball ASs; *(v)* using the resulting PoP locations from our method and the state of the art AS-level connectivity information, through careful case studies we demonstrate that the peering relationships at the edge of the network are highly diverse and complex.

The work described in this chapter has demonstrated the potential of obtaining the geographic footprints of eyeball ASs, which in turn can be used to infer infrastructure-related properties (*e.g.*, PoP locations, AS connectivity) or business-specific features (*e.g.*, serving residential vs. business customers) of these ASs. Doing so by relying solely on measurements at the "edge" of the Internet (*i.e.*, eyeball IP addresses) provides a complementary approach to the more traditional methods for studying the AS-level Internet that exploit exclusively BGP- or traceroute-based measurements. It also suggests a possible fusion of the two approaches whereby the former is augmented with tracerouting capabilities **from** the "edge" and the latter is empowered with performing targeted tracerouting **towards** the edge of the Internet (*i.e.*, eyeballs). Such a combined approach holds the promise to unearth much of what has remained invisible in the AS-level Internet and reveal a maze of real-world peering relationships whose solution will require substantial future research efforts.

In this chapter we presented a new method for capturing the geographical characteristics of ASs and tried to understand the inter-AS peering relationships. The ASs together with the peering relationships that connect them together build the infrastructure of the Internet that carries the traffic associated with different Internet applications. A global picture of the traffic exchanged over each peering relationship essentially depicts a map of Internet traffic exchange. Characteristics of such traffic depends on: *(i)* How the Internet applications demand, or impose traffic on the underlying network, and *(ii)* How the underlying network routes the traffic. In the next chapter, we study the impact of a P2P application overlay, namely Gnutella, on the underlying network, the AS-level underlay.

CHAPTER VII

IMPACT OF P2P OVERLAY ON AS-LEVEL UNDERLAY

Most of the content from this chapter has been adopted from my previously published paper [106] co-authored with Prof. Reza Rejaie and Dr. Walter Willinger. The experimental work is mine and the text has been contributed by myself and the co-authors.

The large volume of traffic associated with Peer-to-Peer (P2P) applications has led to a growing concern among ISPs which need to carry the P2P traffic relayed by their costumers. This concern has led researchers and practitioners to focus on the idea of reducing the volume of external P2P traffic for edge ISPs by localizing the connectivity of the P2P overlay (for recent work, see for example[2, 31]). However, such an approach only deals with the local effect of an overlay on individual edge ASs. Even though the volume of P2P traffic on the Internet is large and growing, assessing the *global* impact of a P2P overlay on the individual ASs in the network, which we call the *AS-level underlay*, remains a challenging problem and is not well understood. This is in part due to the fact that investigating this problem requires a solid understanding of an array of issues in two different domains: *(i)* design and characterization of overlay-based applications, and *(ii)* characterization of AS-level underlay topology and BGP routing in this underlay. Another significant challenge is dealing with inaccurate, missing, or ambiguous information about the AS-level underlay topology, AS relationships and tier properties, and BGP routing policies.

This chapter investigates the problem of assessing the load imposed by a given overlay on the AS-level underlay. We show that assessing this impact requires tackling a number of challenging problems, including *(i)* capturing accurate snapshots of the desired overlay, *(ii)* estimating the load associated with individual overlay connections, and *(iii)* determining the AS-path in the underlay that corresponds to individual overlay connections. Toward this end, this chapter makes two main contributions. First, we present a methodology for assessing the impact of an overlay on the AS-level underlay. Our methodology incorporates a collection of the best known practices for capturing accurate snapshots of a P2P overlay and, more importantly, for determining the AS-path corresponding to each overlay connection. We rely on snapshots of the AS-level Internet topology provided by CAIDA where each link between two ASs is annotated with the relationship between them. Using a BGP simulator called *C-BGP* [99], we perform a detailed simulation of BGP routing over these annotated snapshots of the AS-level underlay to infer the corresponding AS-path for each overlay connection and determine the aggregate load crossing individual ASs. To assess the propagation of overlay traffic through the AS-level hierarchy, we also infer the tier information for individual ASs using the *TierClassify* tool [49].

Second, we illustrate our methodology by characterizing the impact of four snapshots of the *Gnutella* overlay that were captured over four successive years on the AS-level underlay snapshots of the Internet taken on the same dates the Gnutella overlay snapshots were obtained. Our analysis provides valuable insight into how changes in overlay connectivity and underlay topology affect the mapping of load on the AS-level underlay.

The rest of this chapter is organized as follows. In Section 7.1., we further elaborate on the problem of mapping an overlay on the AS-level underlay, describe the challenges involved, and present our methodology. Section 7.2. describes our datasets and presents our characterization of the load imposed by the Gnutella overlay on the corresponding AS-level underlay, spanning a 4-year period.

7.1. The Problem and Our Methodology

Our goal is to map the traffic associated with a P2P overlay to the AS-level underlay. The input to this process is a representation of a P2P overlay structure consisting of the IP addresses (and port numbers) of the participating peers together with their neighbor lists. The output is the aggregate load on all affected ASs and between each pair of affected ASs that have a peering link with one another (in each direction). Our methodology to tackle this problem consists of the following intuitive steps:

1. Capturing the topology of a P2P overlay,
2. Estimating the load on individual connections in the overlay,
3. Inferring the AS-paths associated with individual overlay connections,
4. Determining the aggregate load on each AS and between connected ASs (in each direction separately).

In this section, we discuss the challenges posed by each step, clarify our assumptions, and describe our approach for each step.

7.1.1. Capturing the Overlay Topology

Capturing a snapshot of the overlay topology for a P2P application is feasible if the list of neighbors for individual peers can be obtained. For example, in Gnutella it is possible to query individual peers and retrieve their neighbor lists. Therefore, a Gnutella-specific crawler can be developed to progressively collect this information until a complete snapshot of the overlay is captured.

In our earlier work, we have developed a fast P2P crawler that can capture accurate snapshots of the Gnutella network in a few minutes [131]. Using this crawler, we have captured tens of thousands of snapshots of the Gnutella overlay topology over the past several years. In this study, we use a few of these snapshots for the top-level overlay of Gnutella (an overlay consisting of Gnutella *Ultrapeers*). While other P2P applications such as BitTorrent are responsible for a significantly larger volume of traffic over the Internet than Gnutella and would therefore provide a more relevant P2P system for this study, we are not aware of any reliable technique to capture accurate snapshots of the corresponding overlays. Since accuracy of the overlay topology is important in this study, we focus on Gnutella. However, our methodology is not restricted to this application and can be used with other P2P systems.

7.1.2. Estimating the Load of Individual Overlay Connections

The load of individual overlay connections depends on the subtle interactions between several factors including: *(i)* the number of peers that generate traffic (*i.e.*, sources), the rate and pattern of traffic generation by these peers, and their relative location in the overlay, *(ii)* the topology of the overlay, and *(iii)* the relaying (*i.e.*, routing) strategy at individual peers. Capturing these factors in a single model is a non-trivial task and could be application-specific. For example, the load of individual connections for live P2P video streaming is more or less constant, whereas the load of individual BitTorrent connections may vary significantly over time.

In the absence of any reliable model for per-connection traffic, without loss of generality, we assume in our analysis that all connections of the overlay experience the same average load in both directions. This simplifying assumption allows us to focus on the mapping of the overlay topology on the underlying AS-level topology. If a more reliable model for the load of individual connections is available, it can be easily plugged into our methodology by assigning proper weights (one in each direction) to each connection of the overlay. In this chapter, we simply assume that the weight for all connections in both directions is one.

7.1.3. Inferring AS-Paths for Individual Overlay Connections

For each connection in the overlay, determining the corresponding AS-path in the underlay is clearly the most important and most challenging part of our methodology. We use a popular BGP simulator to determine the AS-path between any given pair of ASs, but note that carefully-designed measurement-based approaches may provide viable alternatives. Our simulation-based method consists of the following steps:

7.1.3.1. Mapping Peers to ASs

We use archived BGP snapshots from RouteViews [93] to map the IP addresses of individual peers to their corresponding ASs that we call edge ASs. Therefore, determining the AS-path for the overlay connection between two peers translates into determining the path between their corresponding edge ASs.

7.1.3.2. Capturing AS-level Topology and Inter-AS Relationships

In this study, we rely on the AS-level topologies provided by CAIDA [20]. These topologies have been widely used in the past, even though more recent work has shown that the provided topologies are missing a significant portion of peering links between lower-tiered ASs [94, 112]. Note that our approach is not tied to using the CAIDA-provided AS-level topologies, and any more complete AS-level topology can be incorporated once it becomes available. To properly simulate BGP routing, we need to determine the AS relationship between connected ASs in the AS-level topology. Toward this end, we use the fact that CAIDA's snapshots of the AS-level topology [20] are annotated with the inferred relationships between each pair of connected ASs. In these snapshots, AS relationships are inferred using the algorithm initially proposed by Gao [48] and extended by Dimitropoulos et al. [41]. This algorithm, mainly based on the concept of "valley-free routing" in BGP (along with some other intuitive assumptions), categorizes the AS relationships into three categories: *(i)* Customer-Provider, *(ii)* Peer-Peer, or *(iii)* Sibling-Sibling.

7.1.3.3. Simulating BGP

We determine the AS-path between any pair of edge ASs that host connected peers in the overlay (*i.e.*, infer the corresponding AS-path) by simulating BGP over the annotated AS-level topology using the *C-BGP* simulator [99]. C-BGP abstracts the AS-level topology as a collection of interconnected routers, where each router represents an AS. It simulates the desired BGP routing policies for each relation between connected ASs. We use a set of intuitive BGP policies for each type of AS relationships that are specified by C-BGP. In particular, these policies *(i)* ensure that the routes through one's customers have the highest preference and those passing through its providers have the lowest preference, and *(ii)* prevent ASs with multiple providers from acting as transit node among their providers. We noticed that some characteristics of CAIDA's annotated AS-level topology, in particular the presence of circular provider-costumer relationships among a group of ASs, prevent our C-BGP simulations to converge with the above policies. To resolve these problems, we systematically change a small number of relationships (*e.g.*, to break a cycle in customer-provider relationships). Further details of this process are described in our related technical report [107]. We select snapshots of both the AS-level topology and the overlay topology of the same dates so as to minimize any potential error due to asynchrony in the snapshots.

Clearly, representing each AS by a single router results in inferring only one AS-path between each pair of ASs. This implies that multiple AS-paths that may exist in practice between two ASs [91] are not accounted for in our simulations. While this assumption simplifies the problem in a way that is not easily quantifiable, we are not aware of any existing technique that can reliably capture and account for this subtle behavior of BGP routing.

7.1.3.4. Assessing AS Tiers

To characterize the propagation of P2P traffic through the AS-level hierarchy, we first need to assess the location of each AS in this hierarchy. We use the "TierClassify" tool [49] to identify the *tier* of each individual AS. The algorithm used in this tool relies mainly on the assumption that all tier-1 ASs should be interconnected with one another. Therefore it tries to find a clique among the ASs with highest degrees. Once the tier-1 clique is identified, the algorithm simply follows provider-customer relationships and classifies other ASs such that each tier n AS can reach the tier-1 clique in $n - 1$ hops.

7.1.3.5. Determining Aggregate Load on and between Individual ASs

Given the corresponding AS-path for each overlay connection, we can easily determine the aggregate load (in terms of the number of connections) that passes through each AS, as well as the transit load (in each direction) between each pair of connected ASs in the topology.

7.2. Effect of the Overlay on the Underlay

In this section, we characterize the effect of a P2P overlay on the AS-level underlay using four snapshots of the Gnutella top-level overlay. We broadly divide ASs into two groups: *Edge ASs* that host peers in an overlay, and *Transit* (or *Core*) *ASs* that provide connectivity between edge ASs. We first describe our datasets (*i.e.*, the snapshots of overlay and the corresponding AS-level underlay topologies), and then we characterize the imposed load on the underlay using the following measures: *(i)* diversity and load on individual AS-paths, *(ii)* load on individual transit ASs, *(iii)* identity and evolution of the top transit ASs, *(iv)* AS-path length, and *(v)* propagation of traffic through the AS-level hierarchy.

7.2.1. Datasets

We use four snapshots of the top-level Gnutella overlay that were collected in four consecutive years starting in 2004. Examining overlay snapshots over time enables us to assess some trends that are associated with the evolution of the AS-level topology.

We use the labels G-xx to refer to the snapshot taken in year 20xx. The left columns of Table 7.1. (labeled "Gnutella snapshots") summarize the capture date, number of peers and edges for these overlay snapshots. The table shows that the population of Gnutella peers in the top-level overlay and their pairwise connections have both increased by $\approx 600\%$ during this four-year period.

We also use daily snapshots of the BGP routing table retrieved from the RouteViews archive collected at the same dates as our overlay snapshots. The middle

TABLE 7.1. Data profile: Gnutella snapshots, BGP snapshots and mapping overlay connections to the underlay. Imp. AS-paths are those with +100 overlay connections.

		Gnutella Snapshots		BGP Snapshots		AS-Paths	
Snapshot	Date	#Peers	#Conn.	#Prefixes	#ASs	#Unique	%Important
G-04	04-11-20	177k	1.46M	165k	18.7k	192k	2.0
G-05	05-08-30	681k	5.83M	185k	20.6k	384k	2.9
G-06	06-08-25	1.0M	8.64M	210k	23.2k	605k	2.8
G-07	07-03-15	1.2M	9.80M	229k	24.9k	684k	2.7

columns in Table 7.1. (labeled "BGP snapshots") give the number of IP prefixes and the total number of ASs in each BGP snapshot. These numbers show that the AS-level topology has also grown significantly during this four-year period.

7.2.2. Diversity and Load on Individual AS-Paths

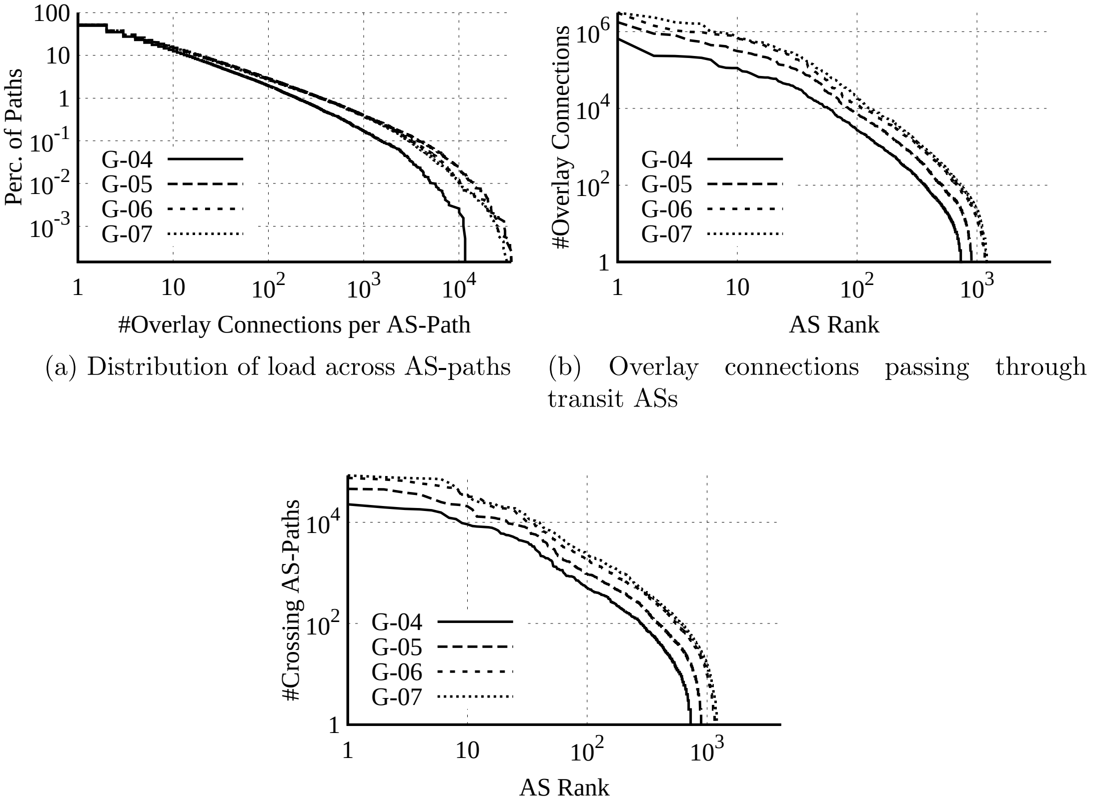
One way to characterize the impact of an overlay on the underlay is to determine the number of unique AS-paths that all overlay connections are mapped on as well as distribution of load among those AS-paths. The right columns of Table 7.1. (labeled "AS-paths") show the number of unique AS-paths for all connections of each overlay along with the percentage of those paths that carry more than 100 overlay connections. The number of unique AS-paths is growing over time but at a lower pace compared to the number of overlay connections. This suggests that there is more similarity in AS-paths among overlay connections as the overlay grows in size over time.

To examine the mapping of overlay connections to AS-paths more closely, Figure 7.1.a depicts the CCDF of the number of overlay connections that map to individual AS-paths in log-log scale for all four overlay snapshots. The skewed shape of these distributions indicates that a small number of AS-paths carry a large fraction of load. For example, whereas around 10% of paths carry more than 10 connections, only 1% of the paths carry more than 200 connections. Interestingly, the distributions of overlay connections that map to AS-paths are very similar across different snapshots despite significant changes in the identity of peers and in the topologies of overlay and underlay.

7.2.3. Observed Load on Individual Transit ASs

Since we assumed that all overlay connections have the same load, we simply quantify the load on each transit AS by the number of overlay connections crossing that AS. Figure 7.1.b depicts the number of overlay connections that cross each transit AS in log-log scale, where ASs are ranked (from high to low) based on their overall observed load. The figure shows that the load on transit ASs is very skewed. A small number of them carry a large volume of traffic while the load on most transit ASs is rather small. Again, we observe that the overall shape of the resulting curves is very similar for all four snapshots, except for the outward shift in the more recent snapshots caused by the increasing size of the overlay over time. This similarity in the skewness of the observed load on transit ASs despite significant changes in the overlay and underlay topologies over time could be due to the dominance of one the following factors: *(i)* the stability over time of the top-10 ASs that host most peers, and *(ii)* the constraint imposed by valley-free routing over the hierarchical structure of the AS-level underlay.

To further investigate the underlying causes for the observed skewed nature of observed load on transit ASs, we examine the distribution of the number of unique AS-paths (associated with overlay connections) that pass through each transit AS in Figure 7.1.c. The shape of this distribution is very similar to Figure 7.1.b, suggesting that the number of crossing connections for individual ASs is primarily determined by the underlay shape and routing rather than connectivity and footprint of the



(a) Distribution of load across AS-paths

(b) Overlay connections passing through transit ASs

(c) AS-paths crossing each transit ASs

FIGURE 7.1. Traffic load distribution

overlay. Figure 7.2. validates this observation by showing a scatterplot of the number of crossing AS-paths (x-axis) and number of overlay connections (y-axis) through each transit AS. This figure essentially relates the previous two distributions and confirms that the observed load on individual transit ASs depends primarily on the number of unique AS-paths crossing those ASs. Note that once the number of cross AS-paths exceeds a certain threshold (a few hundreds), the observed load increases at a much faster pace.

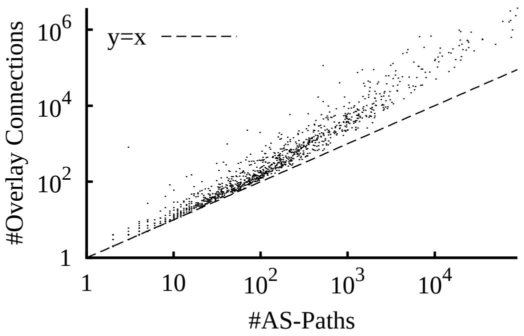


FIGURE 7.2. Scatterplot of number of relevant AS-paths vs. load

7.2.4. Identity and Evolution of Transit ASs

To investigate the observed load by transit ASs from a different angle, we examine and present the identity of the top-10 transit ASs that carry the highest number of crossing overlay connections (and their evolution over time) in Figure 7.3.. For each of the four overlay snapshots, the transit ASs are rank-ordered (highest load first), and the figure depicts their standings in these rank-ordered lists over time. We observe that only four transit ASs (*i.e.*, AT&T, AOL, Level3, and Cogent) remain in the top-10 list across all four snapshots and that the changes in the other transit ASs is more chaotic. This is due to the fact that ranking of transit ASs is affected by a combination of factors including changes in the topology of AS-level underlay, in routing policies, and in the location of peers. Disentangling these different factors and trying to identify the root causes for the observed churn among the top-10 transit ASs over time remains a challenging open problem.

7.2.5. AS-Path Length

One way to quantify the impact of an overlay on the AS-level underlay is to characterize the length of AS-paths for individual overlay connections. Figure 7.4.a

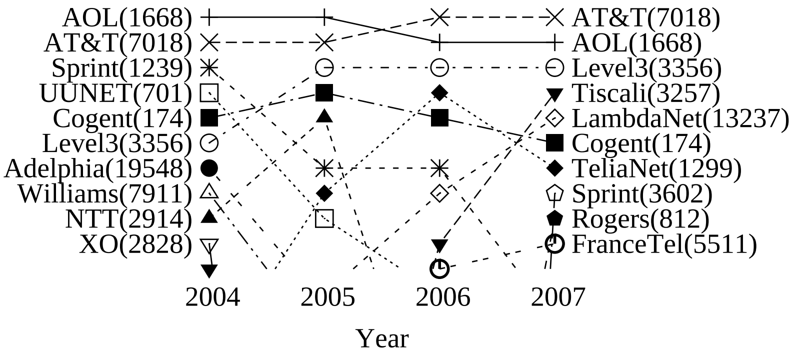
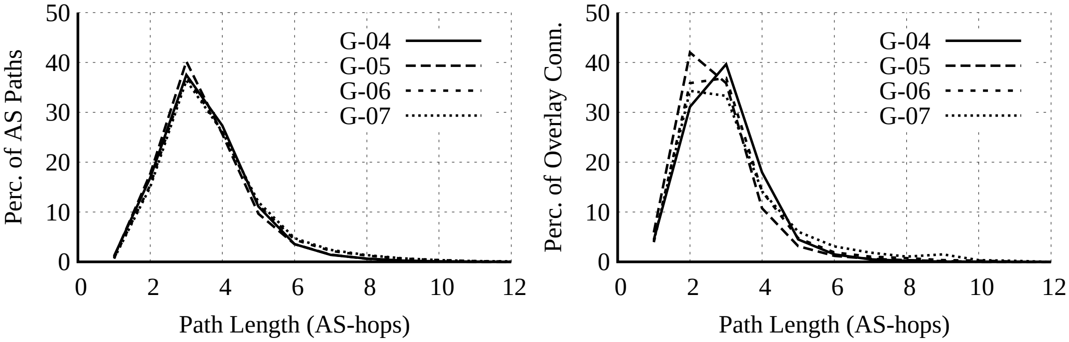


FIGURE 7.3. Identity and evolution of top-10 transit ASs carrying the largest number of overlay connections



(a) Distribution of AS-path length between connected edge ASs

(b) Distribution of AS-path length for all overlay connections

FIGURE 7.4. Distribution of AS-path length

shows the empirical density of the length of all AS-paths between edge ASs for each of the four snapshots. We observe that around 40% of the paths are three AS-hops long, while 80% of the paths in each overlay are at most 4 AS-hops long.

Figure 7.4.b depicts the empirical density of AS-path length across all *overlay connections* for each of the four snapshots. In essence, this plot can be viewed as a *weighted* version of Figure 7.4.a described above where the length of each path is weighted by the number of overlay connections crossing it. The figure shows a very similar pattern across all overlay snapshots despite the changes in the number of peers and their connections. The two figures are very similar, however the average path length across the overlay connections is slightly shorter indicating that a slightly higher fraction of connections are associated with shorter paths. (*e.g.*, for G-07, the average length of all AS-paths is 3.2 hops while the average path length across overlay connections is 3.7 hops.)

7.2.6. Propagation of Traffic through the AS-Level Hierarchy

An interesting way to quantify the load that an overlay imposes on the AS-level underlay is to determine the fraction of load that is propagated upward in the AS-level hierarchy towards the top-tiered ASs. Table 7.2. gives the percentage of paths and percentage of overlay connections whose top AS is a tier-1, tier-2, and tier-3 AS, respectively, in each overlay snapshot. The columns marked "Path" give the percentage of the relevant AS-paths reaching each tier while the columns marked "Conn" represent the percentage of the overlay connections (*i.e.*, aggregate load) reaching each tier. We note that more than half of the paths reach a tier-1 AS, and roughly 40% of the paths peak at a tier-2 AS across all four snapshots.

The percentage of connections that reach a tier-1 AS is even higher than that for paths, indicating that a larger fraction of connections are mapped to these paths. At the same time, a lower percentage of connections reach a tier-2 AS (16% to 37%) compared to paths that peak in tier-2 ASs. Interestingly, the percentage of connections that reach a tier-1 AS decreases over time while the percentage of connections that peak in a tier-2 AS is increasing. A plausible explanation of this trend is the increasing connectivity over time between ASs in the lower tiers which reduces the fraction of connections that have to climb the hierarchy up to tier-1 ASs. A closer examination (not shown here) confirmed that this shift in traffic towards

TABLE 7.2. Percentage of paths/connections reaching each tier of AS hierarchy

	Tier-1		Tier-2		Tier-3	
Snapshot	Path	Conn	Path	Conn	Path	Conn
G-04	51	84	46	16	2.4	0.0
G-05	59	73	38	27	3.0	0.0
G-06	52	64	38	36	10	0.0
G-07	55	63	41	37	3.6	0.1

lower tiers is indeed primarily due to the presence of shortcuts between lower-tier ASs in the AS topology (*e.g.*, more aggressive peering at Internet exchange points over time). In particular, the observed shift has little to do with changes in the overlay topology, mainly because the connectivity of the Gnutella overlay has not become significantly more localized over time.

7.3. Summary

In this chapter, we studied the problem of quantifying the load that a particular overlay imposes on the AS-level underlay. We identified the challenging aspects of this problem and described existing techniques to address each of these aspects. We presented a methodology for mapping the load of an application-level overlay onto the AS-level underlay. We illustrated our methodology with an example of a real-world P2P overlay (*i.e.*, Gnutella).

This chapter makes two main contributions. First, we propose a methodology for capturing the impact of an overlay on the AS-level underlay. The method involves the best known practices for capturing snapshots of a P2P overlay as well as determining the AS-path corresponding to each overlay connection. We use the AS-level snapshots provided by CAIDA in which each AS-AS links are annotated with the type of relationship between the two ASs. Using C-BGP tool, via simulation of BGP routing over the annotated AS-level snapshots, we infer the corresponding AS-paths for each overlay connection and determine the aggregate load crossing individual ASs. Second, using our methodology we characterize the impact of four snapshots of the Gnutella overlay captured during 2004-2007 on the AS-level topology snapshots of the Internet taken on the same dates as the Gnutella overlay snapshots. We characterize the load imposed by these overlays on the corresponding underlay in a number of different ways: (i) observed load on individual AS-paths and its diversity, (ii) observed load on individual transit ASs, (iii) AS-path length, and (iv) the propagation of overlay traffic through the AS-level hierarchy. Our analysis provides valuable insight into how changes in overlay connectivity and underlay topology affect the mapping of load on the AS-level underlay.

From the presented results, we find that the distribution of load across AS-paths is highly skewed such that top 0.001% of the paths carry more than 10,000 overlay connections, the bottom 90% carry less than 10 connections each. We attribute this highly skewed load distribution to the highly hierarchical structure of the AS-level underlay. Although both the overlay and the underlay have grown over 4 years, the distribution patterns remain the same.

Ranking the ASs by the volume of transit traffic, we observe that four of the top-10 transit ASs remain in top-10 during the four years of our study and other ASs are gradually replaced by other ASs. Examining the AS-path length distribution across overlay connections, we notice that the average length is decreasing over the 4-years despite the growth in network size. We also notice that the percentage of overlay connections reaching a tier-1 AS has decreased from 84% in 2004 to 63% in 2007, suggesting that the AS connectivity is becoming more decentralized over time and a larger portion of the traffic gets to destination using peering links without having to go to the top of the AS hierarchy.

While our study contributes to a deeper understanding of the interactions between application-level overlays and the AS-level underlay in today's Internet, a more detailed analysis of the sensitivity of our results to known overlay-specific issues, known underlay-related problems (*e.g.*, incomplete AS graph, ambiguous AS relationships), and known BGP-related difficulties looms as important next step.

CHAPTER VIII

CONCLUSION

This chapter concludes the dissertation by outlining the main contributions of the research presented in each chapter, and by suggesting related future work.

8.1. Contributions

This dissertation has focused on the mutual impacts of the P2P overlay and the AS-level underlay. Towards this end, we presented our work on P2P applications (Chapters III to V), mapping of the AS-level underlay (Chapter VI), and the impact of the P2P overlay on the AS-level underlay (Chapter VII). Below we summarize our main contributions on each of these issues.

8.1.1. Measurement Study on Gnutella Overlay

In Chapter III, we monitored the popular Gnutella P2P overlay by taking frequent snapshots for 15 months using our fast P2P crawler, *Cruiser*. We illustrated how the two-tier overlay has evolved in order to accommodate dramatic changes in user population during the 15 month measurement period. We have explored potential correlations between the evolution of overlay properties and the popularity of different versions of major client programs. Finally, we illustrated the intra-region bias in the connectivity among peers.

Our results illustrate two important points: First, as the Gnutella network has experienced a dramatic increase in user population, the two-tier overlay has repeatedly begun to lose its balance. However, proper modifications in major client software coupled with the rapid upgrade rate of users, has enabled the developers to maintain the overlay's desired properties. Second, despite its random connectivity, the Gnutella overlay exhibits a strong bias towards intra-continent connectivity, especially in continents with smaller user populations. Furthermore, this bias has not changed as the population has quadrupled. In particular, our study of geographic properties of the Gnutella P2P overlay shows dominance of users in English-speaking countries with a strong bias toward intra-region connectivity.

The key contributions of our work include the rigorous measurement of graph theoretic and geographic characteristics of the Gnutella overlay network. Through use of the Cruiser crawler, we have captured much more accurate measurements than previously reported. Our measurement study has also taken more snapshots than other work because of the speed of Cruiser. These measurements will serve as useful baseline measurements for future studies of P2P overlay characteristics.

8.1.2. Sampling of Large-Scale Overlays

In Chapter IV, we proposed a new sampling technique, namely, respondent-driven sampling (RDS) in order to capture the characteristics of large-scale P2P overlays, for which taking accurate full snapshots is not feasible. We explain the technique, describe its theoretical basis, and present a detailed comparison of the proposed technique with a previous sampling technique named Metropolized Random-Walk (MRW), over a variety of synthesized and real world networks.

Our main findings can be summarized as follows: First, RDS outperforms MRW across all scenarios. In particular, RDS exhibits significantly better performance than MRW when the overlay structure exhibits a combination of highly skewed node degrees and highly skewed (local) clustering coefficients. Second, our simulation and empirical evaluations reveal that both the RDS and MRW techniques can accurately estimate key peer properties over dynamic unstructured overlays. Third, our empirical evaluations suggest that the efficiency of the two sampling techniques in practice is lower than in our simulations involving synthetic graphs. We attribute this to our inability to capture accurate reference snapshots.

RDS can be used as a viable tool for sampling large-scale static or dynamic networks on which the researcher has the ability to crawl, *i.e.*, query a node for the list of its neighbors.

8.1.3. P2P Performance Evaluation

In Chapter V, we focused on another aspect of P2P applications which is the most important from the user's perspective: observed performance. Using BitTorrent tracker log files, we extracted several parameters pinpointing the status of participating peers as a group as well as the status of each individual peer. We propose several defining parameters for observed peer performance and then tried to correlate these parameters with several group- and peer-level properties, using different statistical techniques.

Our main contributions in this chapter are the following: *(i)* we present a set of techniques enabling a researcher to capture peer-level and group-level properties of the participating peers in a torrent using the log files normally generated by the trackers; *(ii)* we show that the commonly used technique of "instrumented clients" falls short in accurately characterizing peer-level performance in BitTorrent; and *(iii)* we show that establishing relationships and models between the peer-level performance and other peer- or group-level properties is non-trivial, and no single dominant factor can determine peer-level performance with acceptable statistical significance.

8.1.4. Geographical Mapping of AS-Level Underlay

In Chapter VI, we tackled the problem of AS geography, *i.e.*, inferring an ASs geographical coverage (geo-footprint) and identifying its likely PoP locations. Our approach is based on extensive monitoring of P2P networks and capturing the IP addresses of participating peers, mapping these IP addresses to the ASs where they belong and also finding their geographic locations on the map. This approach, considering P2P users as pins pinpointing provider ASs, is complementary to the traditional BGP- or traceroute-based method of inferring AS-level connectivity due to its focus on the edge of the network, *i.e.*, ASs close to users, rather than at the core of the network.

The main contributions in this chapter are the following: *(i)* we present a methodology for determining the geo-footprint of eyeball ASs by leveraging the geo-locations of their end users; *(ii)* we use the above method to identify the likely PoP locations of an eyeball AS by associating areas with high user concentration with close-by cities in its geo-footprint; *(iii)* we validate our approach using published PoP data from a number of ASs that make this information available on their websites; and *(iv)* using the PoP locations identified by our method and the AS-level connectivity resulting from a state-of-the-art inference approach, we show that the world of peering relationship at the edge of the network is highly diverse and complex. For example, even simple eyeball ASs tend to peer very actively at local and remote IXPs, especially in Europe, and also maintain rich upstream connectivity.

8.1.5. Impact of the P2P Overlay on the AS-Level Underlay

Throughout Chapter VII, we investigated the problem of assessing the load imposed by a given overlay on the AS-level underlay. Towards this end, (i) we captured P2P overlay snapshots; (ii) using an appropriate overlay traffic model, we derived the load associated with individual overlay connections; and (iii) we determined the AS-path corresponding to individual overlay connections and calculated the aggregate associated load on each AS-AS peering link.

The main contributions of this chapter can be summarized as follows. First we propose a methodology for assessing the P2P overlay-underlay impact incorporating a collection of state of the art practices for capturing P2P snapshots and for determining the AS-path corresponding to each overlay connection. Second, we demonstrate our methodology by characterizing the impact of four snapshots of the Gnutella overlay captured over four years on the AS-level snapshots of the Internet taken on the same dates the Gnutella overlay snapshots were obtained.

The main findings from the results presented in this chapter are the following. The distribution of load across AS-paths is highly skewed with the top 0.001% of the paths carrying more than 10,000 overlay connections and the bottom 90% carrying less than 10 connections each. Similar skewness is observed in the distribution of overlay connections across transit ASs. The observed high skewness of load distribution is mainly due to the highly hierarchical structure of the AS-level underlay. We also observe that while four out of the top-10 ASs remain in the top-10 during 2004-2007, the rest of them get replaced by other ASs.

The average AS-path length slightly decreases over the 4-years despite the growth in underlay network size. Another important observation is that the percentage of overlay connections reaching a tier-1 AS has decreased from 84% in 2004 to 63% in 2007, suggesting that the AS-level underlay connectivity is becoming more decentralized over time and a larger portion of the traffic gets to destination using peering links without having to go to the top of the AS hierarchy. This finding is consistent with the observation in Chapter VI that the ASs are becoming more and more inter-connected by peering links between eyeball ASs.

8.2. Future Work

In this section, I present some research problems I am planning to work on in the future. These proposed research ideas are inspired by the works presented in this dissertation.

8.2.1. Traffic Modeling

In Chapter VII, we tried to build a basic model for inter-AS traffic caused by a P2P application. Although P2P traffic is important, it is only part of the Internet traffic. In order to build a realistic traffic matrix, we will need to identify data centers, content providers and CDNs. A basic idea is to identify the IP addresses of the most popular data sources and then map them into respective AS numbers. However, capturing the IP addresses of the real servers sourcing the traffic is challenging. Popular content providers usually offer their services via Content Distribution Networks (CDNs) in which the client is automatically redirected to the closest server. In such cases, in order to detect the real data sources in a global scale, a researcher needs to have a large number of vantage points distributed across the world. After detecting the data sources, the researcher may assume a simplistic model in which traffic from all data sources flows towards all eyeball ASs in proportion to their user base. More sophisticated models may also consider time of day effects, etc.

The BGP-based method we have used in Chapter VII for detecting AS-paths may be complemented with other sources of AS connectivity information such as traceroute data sets. The resulting traffic matrices, may be analyzed and presented in different useful ways including complex network visualization techniques.

8.2.2. Towards a Directory of Autonomous Systems

In Chapter VI, in order to validate the results of our PoP locating system, we used manual data collection to gather PoP lists for 50 ISPs from their websites. Although time consuming, it was a successful attempt at gathering data on ASs from authoritative sources. This is an important step towards building the first public AS directory over the Internet. Researchers may utilize additional methods including company websites and e-mail communication to gather a variety of information on each individual AS. The information may include PoP locations, inter-AS links and their types and locations, number of customers, areas where services are provided, etc. Such a data base will be very valuable for users, researchers as well as ISPs.

Considering the large number of active ASs, this may seem a prohibitively large amount of work. However, we should notice that once the data is gathered for tier-1 and popular tier-2 ASs, the database will become standardized and widely used urging the rest of the ASs to submit their information to be included on a voluntary basis.

8.2.3. Effect of P2P Traffic on AS Connectivity

According to the traditional Internet usage models, almost all the traffic flows from data centers towards the users. Due to this pattern, the ISPs had little incentive to have peering relationships with one another since there is little traffic demand between ISPs (*i.e.*, between users). Based on the traffic model we proposed in Chapter VII, P2P applications impose a large traffic demand among end-users.

According to this model, which is the result of the widespread use of P2P applications, ISPs may have reasonable economical and technical incentives to connect to one another and exchange traffic to save on costly traffic towards/from their upstream providers.

Our future work could include a study similar to Chapter VII perhaps with more focus on a group of ISPs in a country or a region to derive a traffic demand matrix among ISPs. Using a basic traffic cost model, we could calculate the monetary savings that the ISPs will accrue if they establish peering relationships. Such peering links may also lead to establishing Internet exchange points to lower the costs.

8.2.4. Modeling AS Connectivity

Besides technical factors, ASs primarily follow business incentives when deciding to establish peering links to one-another. ISPs may save on the costly traffic sent to and received from their upstream providers, by establishing peering connections among each other. From an economical point of view, the decision should be made when the savings and benefits of having the link surpasses the costs of establishing and maintaining it. It is reasonable to assume that the relative geographical locations of a pair of ISPs play an important role in such a decision. If the two ISPs have close-by PoPs, or if they are already present in the same location, then the initial cost of establishing a peering link should be very small. Otherwise, they will have to incur data transport charges.

Another important factor can be the size of each AS, both in terms of customer base, and service zone. This is also the determining factor in defining the type of the business agreement governing the traffic exchange. If the ISPs are roughly of the same size and class, the traffic exchange is usually a non-paid peering; otherwise, the smaller ISP becomes a customer of the larger, and pays for the traffic exchange.

Based on this hypothesis, a researcher may derive a model for AS connectivity in which the geographical parameters of the ASs (e.g. PoP locations), as well as their customer base (number of direct and indirect users) are used as input. The model will then calculate, for each pair of ASs, the connectivity parameters such as probability (or value) of having a link, and the type of traffic exchange agreement. Such a system will be very useful for ASs to evaluate and optimize their current and potential inter-AS links including customer-provider and peering links.

REFERENCES CITED

[1] AFRINIC. African Network Information Center. `http://www.afrinic.net`, 2009.

[2] V. Aggarwal, A. Feldmann, and C. Schneideler. Can ISPs and P2P Systems Co-operate for Improved Performance? *ACM Computer Communication Review*, 37(3):29–40, July 2007.

[3] R. Albert, H. Jeong, and A.L. Barabasi. Error and Attack Tolerance of Complex Networks. *Nature*, 406(6794):378–382, July 2000.

[4] D. Alderson, J. Doyle, W. Willinger, and R. Govindan. Toward an Optimization-Driven Framework for Designing and Generating Realistic Internet Topologies. In Proc. *ACM HotNets-I*, 2002.

[5] D. Alderson, L. Li, W. Willinger, and J. C. Doyle. Understanding Internet Topology: Principles, Models, and Validation. *IEEE/ACM Transactions on Networking*, 13(6):1205–1218, 2005.

[6] Alexa. Alexa the Web Information Company. `http://www.alexa.com`, 2009.

[7] APNIC. Asia Pacific Network Information Centre. `http://www.apnic.net`, 2009.

[8] ARIN. American Registry for Internet Numbers. `http://www.arin.net`, 2009.

[9] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding Traceroute Anomalies with Paris Traceroute. In Proc. *ACM Internet Measurement Conference*, October 2006.

[10] B. Augustin, B. Krishnamurthy, and W. Willinger. Ixps: Mapped? In Proc. *ACM Internet Measurement Conference*, November 2009.

[11] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In Proc. *ACM SIGCOMM*, August 2002.

[12] A.L. Barabasi and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, October 1999.

[13] A.L. Barabasi, Z. Dezso, E. Ravasz, S.H. Yook, and Z. Oltvai. Scale-free and Hierarchical Structures in Complex Networks. In Proc. *Seventh Granada Lectures*, 2002.

[14] P. Barford, A. Bestavros, J. Byers, and M. Crovella. On the Marginal Utility of Network Topology Measurements. In Proc. *ACM Internet Measurement Workshop*, 2001.

[15] A. Bharambe, C. Herley, and V. Padmanabhan. Analyzing and Improving a BitTorrent Network's Performance Mechanisms. In Proc. *IEEE INFOCOM*, April 2006.

[16] G. Bildson. CTO, LimeWire LLC. Personal correspondence with Daniel Stutzbach, November 2005.

[17] B. Bollobás. A Probabilistic Proof of an Asymptotic Formula for the Number of Labelled Regular Graphs. *European Journal of Combinatorics*, 1:311–316, 1980.

[18] Z. Botev, J. Grotowski, and D. Kroese. Kernel Density Estimation via Diffusion. *Annals of Statistics*, 2010.

[19] P. A. Branch, A. Heyde, and G. J. Armitage. Rapid Identification of Skype Traffic. In Proc. *ACM NOSSDAV*, 2009.

[20] CAIDA. Cooperative Association for Internet Data Analysis. `http://www.caida.org`, 2008.

[21] CAIDA. Cooperative Association for Internet Data Analysis. Skitter Project. `http://www.caida.org/tools/measurement/skitter/`, 2008.

[22] K. Calvert, M. Doar, A. Nexion, and E. Zegura. Modeling Internet Topology. *IEEE Transactions on Communications*, pages 160–163, December 1997.

[23] M. Castro, P. Druschel, A.M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-Bandwidth Content Distribution In A Cooperative Environment. In Proc. *International Workshop on Peer-to-Peer Systems*, 2003.

[24] H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. Towards Capturing Representative AS-level Internet Topologies. *Computer Networks Journal*, 44(6):737–755, 2004.

[25] H. Chang, S. Jamin, and W. Willinger. Inferring AS-level Internet Topology from Router-level Path Traces. In Proc. *SPIE ITCom*, 2001.

[26] H. Chang, S. Jamin, and W. Willinger. Internet Connectivity at the AS-level: An Optimization-driven Modeling Approach. In Proc. *ACM SIGCOMM Workshop on MoMeTools*, pages 33–46, August 2003.

[27] K. Chen, D. Choffnes, R. Potharaju, Y. Chen, F. Bustamante, D. Pei, and Y. Zhao. Where the Sidewalk Ends: Extending the Internet AS Graph Using Traceroutes From P2P Users. In Proc. *ACM CoNEXT*, December 2009.

[28] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. The Origin of Power-Laws in Internet Topologies Revisited. In Proc. *IEEE INFOCOM*, June 2002.

[29] Cheswick.com Website. Internet Mapping Project. http://www.cheswick.com/ches/map, 2000.

[30] S. Chib and E. Greenberg. Understanding the Metropolis–Hastings Algorithm. *The Americian Statistician*, 49(4):327–335, November 1995.

[31] D. R. Choffnes and F. E. Bustamante. Taming The Torrent: A Practical Approach To Reducing Cross-ISP Traffic In P2P Systems. In Proc. *ACM SIGCOMM*, August 2008.

[32] J. Chu, K. Labonte, and B. N. Levine. Availability and Locality Measurements of Peer-to-Peer File Systems. In Proc. *ITCom: Scalability and Traffic Control in IP Networks II Conferences*, July 2002.

[33] Y. H. Chu, S. G. Rao, S. Seshan, and H. Zhang. A Case for End System Multicast. *IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast*, 20(8), 2002.

[34] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. *Lecture Notes in Computer Science*, 2000.

[35] M. Coates, A. Hero, R. Nowak, and B. Yu. Internet Tomography. *IEEE Signal Processing Magazine*, May 2002.

[36] B. Cohen. Incentives Build Robustness in BitTorrent. In Proc. *First Workshop on Economics of Peer-to-Peer Systems*, May 2003.

[37] Federal Communications Commission. Commission Orders COMCAST to End Discriminatory Network Management Practices. http://hraunfoss.fcc.gov/edocs_public/attachmatch/DOC-284286A1.pdf, August 2008.

[38] P. Crowley. On the Relative Importance of P2P Peer Selection. Internet Draft. http://tools.ietf.org/html/draft-crowley-alto-importance-00, July 2008.

[39] A. Dhamdhere and C. Dovrolis. An Agent-based Model For The Evolution Of The Internet Ecosystem. In Proc. *COMmunication Systems And NETworks*, July 2009.

[40] X. Dimitropoulos. Revealing the Autonomous System Taxonomy: The Machine Learning Approach. In Proc. *Passive and Active Measurements Conference*, March 2006.

[41] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, K. Claffy, and G. Riley. AS Relationships: Inference and Validation. *ACM Computer Communication Review*, 37(1):29–40, 2007.

[42] M. Doar. A Better Model for Generating Test Networks. In Proc. *Global Telecommunications Conference*, London, UK, November 1996.

[43] J. C. Doyle, D. Alderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger. The "Robust Yet Fragile" Nature Of The Internet. *Proceedings of the National Academy of Sciences*, 102(41):14497–1452, 2005.

[44] A. Fabrikant, E. Koutsoupias, and C. H. Papadimitrio. Heuristically Optimized Trade-offs: A New Paradigm For Power Laws In The Internet. In Proc. *ICALP*, 2002.

[45] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. In Proc. *ACM SIGCOMM*, 1999.

[46] P. Faratin, D. Clark, P. Gilmore, and A. Berger. Complexity of Internet Interconnections: Technology, Incentives and Implications for Policy. In Proc. *Telecommunications Policy Research Conference*, September 2007.

[47] A. Fisk. Dynamic Query Protocol. `http://www.the-gdf.org/wiki/index.php?title=Dynamic_Query_Protocol`, 2006.

[48] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9:733–745, 2000.

[49] Z. Ge, D. R. Figueiredo, S. Jaiswal, and L. Gao. On the Hierarchical Structure of the Logical Internet Graph. In Proc. *SPIE ITCom*, November 2001.

[50] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley. Modeling Peer-Peer File Sharing Systems. In Proc. *IEEE INFOCOM*, 2003.

[51] S. Goel and M. J. Sagalnik. Respondent-driven Sampling as Markov Chain Monte Carlo. *Statistics in Medicine*, 28(17):2202–2229, July 2009.

[52] R. Govindan and A. Reddy. An Analysis of Internet Inter-Domain Topology and Route Stability. In Proc. *IEEE INFOCOM*, pages 850–857, 1997.

[53] R. Govindan and H. Tangmunarunkit. Heuristics for Internet Map Discovery. In Proc. *IEEE INFOCOM*, April 2000.

[54] Wand Network Research Group. Scamper. `http://www.wand.net.nz/scamper`, 2009.

[55] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. *ACM SIGOPS Operating Systems Review*, 37(5):314–329, December 2003.

[56] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating Latency between Arbitrary Internet End Hosts. In Proc. *ACM Internet Measurement Workshop*, November 2002.

[57] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, Analysis, and Modeling of BitTorrent-like Systems. In Proc. *ACM Internet Measurement Conference*, October 2005.

[58] M. Hansen and W. Hurwitz. On the Theory of Sampling from Finite Populations. *Annals of Mathematical Statistics*, 14(4):333–362, 1943.

[59] N. J. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman. SkipNet: A Scalable Overlay Network with Practical Locality Properties. In Proc. *USENIX Symposium on Internet Technologies and Systems*, 2003.

[60] W. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57:97–109, 1970.

[61] Y. He, G. Siganos, M. Faloutsos, and S. V. Krishnamurthy. A Systematic Framework for Unearthing the Missing Links: Measurements and Impact. In Proc. *NSDI*, April 2007.

[62] D. Heckathorn. Respondent-driven Sampling II: Deriving Valid Population Estimates from Chain-Referral Samples of Hidden Populations. *Social Problems*, 49(1):11–34, 2002.

[63] Hexasoft. IP2Location. `http://www.ip2location.com/aboutus.aspx`, 2009.

[64] CAIDA Cooperative Association for Internet Data Analysis. Archipelago Measurement Infrastructure. `http://www.caida.org/projects/ark`, 2009.

[65] Information Science Institute, University of Southern California. The Network Simulator - ns-2. `http://www.isi.edu/nsnam/ns`, 2009.

[66] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. A. Felber, A. A. Hamra, and L. Garces-Erice. Dissecting BitTorrent: Five Months in a Torrent's Lifetime. In Proc. *Passive and Active Measurement Workshop*, April 2004.

[67] V. Jacobson. traceroute, 1989.

[68] J. Jannotti, D. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole Jr. Overcast: Reliable Multicasting with an Overlay Network. In Proc. *OSDI*, October 2000.

[69] M. Jovanovic, F. Annexstein, and K. Berman. Modeling Peer-to-Peer Network Topologies through “Small-World” Models and Power Laws. In Proc. *TELFOR*, November 2001.

[70] T. Karagiannis, A. Broido, M. Faloutsos, and kc claffy. Transport Layer Identification of P2P Traffic. In Proc. *International Measurement Conference*, October 2004.

[71] T. Karagiannis, P. Rodriguez, and K. Papagiannaki. Should Internet Service Providers Fear Peer-Assisted Content Distribution? In Proc. *ACM Internet Measurement Conference*, pages 63–76, October 2005.

[72] Y. Kim and K. Chon. Scalable and Topologically-aware Application-layer Multicast. In Proc. *IEEE GLOBECOM*, November 2004.

[73] LACNIC. Latin American and Caribbean Internet Addresses Registry `http://www.lacnic.net`, 2009.

[74] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie. Sampling Biases in IP Topology Measurements. In Proc. *IEEE INFOCOM*, 2003.

[75] A. Legout, G. Urvoy-Keller, and P. Michiardi. Rarest First and Choke Algorithms Are Enough. In Proc. *ACM Internet Measurement Conference*, October 2006.

[76] L. Li, D. Alderson, W. Willinger, and J. C. Doyle. A First-Principles Approach to Understanding the Internet’s Router-Level Topology. In Proc. *ACM SIGCOMM*, pages 3–14, 2004.

[77] M. Li, W. Lee, and A. Sivasubramaniam. Semantic Small World: An Overlay Network for Peer-to-Peer Search. In Proc. *International Conference on Network Protocols*, October 2004.

[78] L. Lovász. Random Walks On Graphs: A Survey. *Combinatorics: Paul Erdös is Eighty*, 2:1–46, 1993.

[79] M. Luckie, Y. Hyun, and B. Huffaker. Traceroute Probe Method and Forward IP Path Inference. In Proc. *ACM Internet Measurement Conference*, 2008.

[80] R. Ma, D. Chiu, J. Lui, V. Misra, and D. Rubenstein. Interconnecting Eyeballs to Content: A Shapley Value Perspective on ISP Peering and Settlement. In Proc. *ACM SIGCOMM workshop on Economics of networked systems*, August 2008.

[81] R. Ma, D. Chiu, J. Lui, V. Misra, and D. Rubenstein. On Cooperative Settlement Between Content, Transit and Eyeball Internet Service Providers. In Proc. *ACM CoNEXT*, December 2008.

[82] N. Magharei and R. Rejaie. PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming. In Proc. *IEEE INFOCOM*, pages 1415–1423, May 2007.

[83] P. Mahadevan, C. Hubble, B. Huffaker, D. Krioukov, and A. Vahdat. Orbis: Rescaling Degree Correlations to Generate Annotated Internet Topologies. In Proc. *ACM SIGCOMM*, August 2007.

[84] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat. Systematic Topology Analysis and Generation Using Degree Correlations,. In Proc. *ACM SIGCOMM*, September 2006.

[85] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, and X. Dimitropoulos. The Internet AS-Level Topology: Three Data Sources and One Definitive Metric. *ACM Computer Communication Review*, 36(1):17–26, 2006.

[86] MaxMind. GeoIP. `http://www.maxmind.com/app/ip-location`, 2006.

[87] P. Maymounkov and D. Mazieres. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. In Proc. *International Workshop on Peer-to-Peer Systems*, 2002.

[88] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation,. In Proc. *MASCOTS*, pages 346–353, August 2001.

[89] A. Medina, I. Matta, and J. Byers. On the Origin of Power Laws in Internet Topologies. *ACM Computer Communication Review*, 2000.

[90] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[91] W. Muhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-topology Model That Captures Route Diversity. *ACM Computer Communication Review*, 36(4):195–206, October 2006.

[92] W. B. Norton. The Evolution of the U.S. Internet Peering Ecosystem. Talk in NANOG 31 `http://www.nanog.org/meetings/nanog31/presentations/nortonslides.pdf`, 2003.

[93] University of Oregon. RouteViews Project. `http://www.routeviews.org/`, 2008.

[94] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. In Search of the Elusive Ground Truth: The Internet's AS-level Connectivity Structure. In Proc. *ACM SIGMETRICS*, 2008.

[95] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. The (In)Completeness of the Observed Internet AS-level Structure. *IEEE/ACM Transactions on Networking*, 18(1):109 –122, February 2010.

[96] V. N. Padmanabhan and K. Sripanidkulchai. The Case for Cooperative Networking. In Proc. *International Workshop on Peer-to-Peer Systems*, 2002.

[97] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips. The Bittorrent P2P File-sharing System: Measurements and Analysis. In Proc. *International Workshop on Peer-to-Peer Systems (IPTPS)*, February 2005.

[98] D. Qiu and R. Srikant. Modeling and Performance Analysis of Bit Torrent-Like Peer-to-Peer Networks. In Proc. *ACM SIGCOMM*, 2004.

[99] B. Quoitin and S. Uhlig. Modeling the Routing of an Autonomous System with C-BGP. *IEEE Network*, 19(6), November 2005.

[100] Merit RADB. Routing Assets Database. `http://www.radb.net`, 2009.

[101] A. Rasti, R. Rejaie, N. Duffield, D. Stutzbach, and W. Willinger. Evaluating Sampling Techniques for Large Dynamic Graphs. Technical report CIS-TR-08-01, Department of Computer and Information Science, University of Oregon, March 2008. `http://mirage.cs.uoregon.edu/pub/tr08-01.pdf`.

[102] A. Rasti, D. Stutzbach, and R. Rejaie. On the Long-term Evolution of the Two-Tier Gnutella Overlay. In Proc. *Global Internet Symposium*, April 2006.

[103] A. Rasti, M. Torkjazi, R. Rejaie, N. Duffield, W. Willinger, and D. Stutzbach. Respondent-driven Sampling for Characterizing Unstructured Overlays. In Proc. *IEEE INFOCOM Mini-conference*, 2009.

[104] A. H. Rasti, N. Magharei, R. Rejaie, and W. Willinger. Eyeball ASes: From Geography to Connectivity. In Proc. *ACM Internet Measurement Conference*, 2010.

[105] A. H. Rasti and R. Rejaie. Understanding Peer-Level Performance in BitTorrent: A Measurement Study. In Proc. *International Conference on Computer Communications and Networks*, August 2007.

[106] A. H. Rasti, R. Rejaie, and W. Willinger. Characterizing the Global Impact of P2P Overlays on the AS-Level Underlay. In Proc. *Passive and Active Measurement Conference*, April 2010.

[107] A. H. Rasti, R. Rejaie, and W. Willinger. Characterizing the Global Impact of the P2P Overlay on the AS-Level Underlay. Technical Report CIS-TR-10-01, Department of Computer and Information Science, University of Oregon, January 2010. `http://mirage.cs.uoregon.edu/pub/tr10-01.pdf`

[108] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In Proc. *ACM SIGCOMM*, 2001.

[109] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-Aware Overlay Construction and Server Selection. In Proc. *IEEE INFOCOM*, June 2002.

[110] RIPE NCC. European IP Networks. `http://www.ripe.net`, 2009.

[111] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design. *IEEE Internet Computing Journal*, 6(1), 2002.

[112] M. Roughan, S. J. Tuke, and O. Maennel. Bigfoot, Sasquatch, the Yeti and Other Missing Links: What We Don't Know About the AS Graph. In Proc. *ACM Internet Measurement Conference*, October 2008.

[113] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-To-Peer Systems. In Proc. *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, November 2001.

[114] M. Salganik and D. Heckathorn. Sampling and Estimation in Hidden Populations Using Respondent-driven Sampling. *Sociological Methodology*, 34:193–239, 2004.

[115] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In Proc. *SPIE/ACM MMCN*, January 2002.

[116] S. Saroiu, P. K. Gummadi, and S. D. Gribble. Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts. *Multimedia Systems Journal*, 9(2):170–184, August 2003.

[117] J. Seedorf and E. Burger. Application-Layer Traffic Optimization (ALTO) Problem Statement. RFC 5693, `http://tools.ietf.org/html/rfc5693`, 2009.

[118] S. Sen and J. Wang. Analyzing Peer-To-Peer Traffic Across Large Networks. *IEEE/ACM Transactions on Networking*, 12(2):219–232, April 2004.

[119] S. Shalunov, R. Penno, and R. Woundy. ALTO Information Export Service. Internet Draft, October 2008.

[120] Y. Shavitt. The DIMES Project. `http://www.netdimes.org`, 2010.

[121] Y. Shavitt and N. Zilberman. A Structural Approach for PoP Geo-Location. In Proc. *IEEE Workshop on Network Science For Communication Networks*, March 2010.

[122] A. Singla and C. Rohrs. Ultrapeers: Another Step Towards Gnutella Scalability. Gnutella Developer's Forum, November 2002.

[123] S. Siwpersad, B. Gueye, and S. Uhlig. Assessing the Geographic Resolution of Exhaustive Tabulation for Geolocating Internet Hosts. In Proc. *Passive and Active Measurement Conference*, April 2008.

[124] Slyck.com Website. `http://www.slyck.com`, 2005.

[125] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP Topologies with Rocketfuel. In Proc. *ACM SIGCOMM*, 2002.

[126] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications. *IEEE/ACM Transactions on Networking*, 2002.

[127] D. Stutzbach and R. Rejaie. Capturing Accurate Snapshots of the Gnutella Network. In Proc. *Global Internet Symposium*, pages 127–132, March 2005.

[128] D. Stutzbach and R. Rejaie. Evaluating the Accuracy of Captured Snapshots by Peer-to-Peer Crawlers. In Proc. *Passive and Active Measurement Workshop*, Extended Abstract, pages 353–357, March 2005.

[129] D. Stutzbach and R. Rejaie. Understanding Churn in Peer-to-Peer Networks. In Proc. *ACM Internet Measurement Conference*, October 2006.

[130] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger. On Unbiased Sampling for Unstructured Peer-to-Peer Networks. *IEEE/ACM Transactions on Networking*, 2008.

[131] D. Stutzbach, R. Rejaie, and S. Sen. Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems. In Proc. *ACM Internet Measurement Conference*, pages 49–62, October 2005.

[132] D. Stutzbach, R. Rejaie, and S. Sen. Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems. *IEEE/ACM Transactions on Networking*, 16(2), April 2008.

[133] L. Subramanian, S. Agarwal, J. Rexford, and Y. H. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. In Proc. *IEEE INFOCOM*, 2002.

[134] K. Suh, D. Figueiredo, J. F. Kurose, and D. Towsley. Characterizing and Detecting Skype-Relayed Traffic. In Proc. *IEEE INFOCOM*, April 2006.

[135] D. J. Watts. Six Degrees: The Science of a Connected Age. ACM Press, 2003.

[136] J. Xia and L. Gao. On the Evaluation of AS Relationship Inferences. In Proc. *IEEE GLOBECOM*, November 2004.

[137] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz. P4P: Provider Portal for Applications. In Proc. *ACM SIGCOMM*, 2008.

[138] S.H. Yook, H. Jeong, and A.L. Barabasi. Modeling the Internet's Large-scale Topology. *Proceedings of the National Academy of Sciences*, 99(21):13382–13386, October 2002.

[139] B. Zhang and R. Liu. Collecting the Internet AS-level Topology. *ACM Computer Communication Review*, 35:53–61, 2005.

[140] X. Zhang, J. Liu, and T. shing Peter Yum. Coolstreaming/Donet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming. In Proc. *IEEE INFOCOM*, March 2005.

[141] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A Resilient Global-Scale Overlay for Service Deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, January 2004.

[142] L. Zou and M. Ammar. A File-Centric Model for Peer-to-Peer File Sharing Systems. In Proc. *International Conference on Network Protocols*, November 2003.