UNIVERSITY OF CALIFORNIA
RIVERSIDE

Measuring and Modeling Applications for Content Distribution in the Internet

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Anirban Banerjee

December 2008

Dissertation Committee:
Dr. Laxmi Bhuyan, Chairperson
Dr. Mart Molle
Dr. Srikanth Krishnamurthy

The Dissertation of Anirban Banerjee is approved:

Committee Chairperson

University of California, Riverside

Acknowledgements

I would like to profusely thank my advisor Dr. Michalis Faloutsos for his patience and guidance. Helping me to improve the innumerable ”fields of devastation” was not an easy job! His energy and optimism is infectious and I am honored to have had the opportunity to work with him. I would also like to thank Dr. Laxmi Bhuyan for his guidance. Furthermore, I would like to thank Dr. Mart Molle who has been one of my favorite people to bounce ideas with. His ability to dissect problems has always encouraged me. Additionally, I would like to thank Dr. Srikanth Krishnamurthy for helping me out despite his busy schedule.

I owe unmeasurable gratitude to my parents, Ashok and Aditi Banerjee who have sacrificed innumerable times to help me along. Thanks baba and ma!

I would like to thank my ”foveri” girlfriend, Nomiki, for motivating me with her effervescent personality.

I would also like to thank my ”awesome” set of friends I have met at UCR, Abhishek, Arindam, Hiya, Indranil, Dhiman, Yihua, Satya, Rupam, Arunava, Santanu, Ece, Marios, Yannis, Aggelos, Nicholas, Doxa, Sherry, Smruti, Archana, Yordanos, Ilker, Serdar, Mostafa, Anna, Tatev, Anne, Joanna, Toi and so many other wonderful people. I must thank Dr. Taradas Bandyopadhyay, Bharati Bandyopadhyay, Trisha and Esha for all the affection they have shown me.

The systems staff, in particular Victor Hill, has been unbelievable with his help. The office staff, Terri, Amy, Madie, Jackie, Lilian, Vanoohi and all the others have been so helpful in every way. I would like to thank the folks at the Internal Education Center, Kelly and Deja.

Finally, I would like to thank Dr. Christian Sheldon, Dr. Neal Young, Dr. Stefano Lonardi and Dr. Walid Najjar for allowing me to discuss ideas with them.

Man is made by his belief. As he believes, so he is. - Bhagavad Gita

ABSTRACT OF THE DISSERTATION

Measuring and Modeling Applications for Content Distribution in the Internet

by

Anirban Banerjee

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, December 2008
Dr. Laxmi Bhuyan, Chairperson

The focus of this dissertation is on measuring, analyzing and modeling emerging applications in the Internet. Specifically, we concentrate on understanding the internals of content distribution paradigms such as Peer-to-Peer (P2P) systems and podcasts. This dissertation consists of three main thrusts which we describe below.

P2P streams have been reported to constitute nearly 61% of all upstream traffic. P2P streams are used for disseminating content ranging from video programs to linux images. This everpresent ubiquity of P2P networks has also allowed them to be used for sharing copyrighted material. This has resulted in organizations like the RIAA, taking legal action against file-sharers. As a result P2P users have employed defenses against being monitored by such organizations. We have found that a little caution pays off a lot, since there is a 100% probability of a naive P2P user being monitored when accessing these networks.

Further, as a case study, we present a comprehensive study about eDonkey, a popular P2P network. We identify the limitations of current approaches to measure P2P networks. Additionally, we find that P2P flows traverse through the Internet quite differently than http flows. Based on this, we present metrics useful for distinguishing P2P traffic from other forms of traditional content distribution in the Internet.

Finally, podcasts, a relatively new content distribution mechanism is expected to garner an audience of nearly 56 million subscribers by 2010. Measuring and modeling podcasts remains an open problem despite the significance that has been gained by this application. This form of content distribution is best described as a push based mechanism, which is different from traditional http based content distribution. We measure podcast streams, analyze them and develop a traffic generator, SimPod, for simulation purposes.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Content distribution in the Internet has been revolutionized with the advent of P2P networks and podcasts. These emerging technologies have enabled Internet users to share and disseminate information in the form of videos, audio files, and various other formats at an unprecedented level. In this thesis we discuss these two disruptive technologies which have had a profound effect on the digital landscape.

P2P networks can be defined as cooperating groups of computers which work together to share resources such as data files. Each peer is an individual computer who contributes a part of or a complete file to the resource pool shared across the network. All peers work with each other based on specific protocols to gather information from a multitude of peers to finally obtain the data each one needs.

Podcasts are a relatively recent phenomenon. Internet users who want to disseminate audio files, video files documenting their thoughts and experiences can turn them into podcasts.

Users can subscribe to these podcast streams and download the files when available. This mode of content distribution has exploded in popularity because of its simplicity and ease of setup. A publisher of information does not need an expensive content distribution setup.

We focus on P2P networks during the first half of the thesis followed by an analysis of podcasts in the second. In more detail, our work consists of three thrusts: (a) analyzing privacy issues in P2P networks (b) measuring and modeling spatial characteristics of P2P networks and users and (c) an analysis of podcast content distribution and the development of a comprehensive podcast model.

1.1 Analyzing Privacy Issues in P2P Networks

The increasing trend of Internet users employing P2P networks for downloading media has been well documented [18, 21, 5]. Data-streams attributed to file sharing via P2P networks have been reported to constitute a major portion of ISP traffic in the Internet. P2P streams have been reported to constitute nearly 61% of all upstream traffic [5]. Since P2P networks are widely employed for sharing copyrighted material, media companies have taken legal recourses to try and stem the tide of this immense problem. This has led to a digital war of sorts, the media companies attempt to poison or stop file sharers from using P2P networks, while the users of such networks attempt to devise constantly changing strategies and software solution to outwit the other side. On one hand the RIAA [27] and P2P monitoring companies [3] attempt to restrict the operation of P2P networks by legal and technological

[6] means, while on the other hand P2P file sharers attempt to use **blocklists** [4, 14] to avoid running into honeypot servers which log activities of file sharers downloading copyrighted material. We believe our work is the first of its kind to document the effect of this kind of digital war.

In an effort to legally prosecute P2P users, the RIAA and MPAA have reportedly started to create decoy users: they participate in P2P networks in order to identify illegal sharing of content. This has reportedly scared some users who are afraid of being caught and prosecuted. The question we would like to answer is how prevalent is this phenomenon: how likely is it that a user will run into such a "fake user" and thus run the risk of a lawsuit? The first challenge is identifying these "fake users". We collect this information from a number of free open source software projects which are trying to identify such addresses by forming the, so called, blocklists. The second challenge is to quantify the probability of a user contacting such a fake user by conducting a large scale experiment in order to obtain reliable statistics. Using Planetlab, we conduct active measurements, spanning a period of 90 days, from January to March 2006, spread over 3 continents. Analyzing over 100 GB of TCP header data, we quantify the probability of a P2P user contacting fake users. We observe that the probability of peers contacting entities in these lists is nearly 100%. In fact, 12 to 17% of all distinct IPs contacted by any node were listed on blocklists. Interestingly, a little caution can have significant effect: the top five most prevalent blocklisted IP ranges contribute to nearly 94% of all blocklisted IPs we ran into. Avoiding these can reduce the probability of a user being tracked to about 1%. In addition, we examine the identity of these blocklisted IPs. Interestingly, less than 0.5% of all unique IPs contacted, belong explicitly to media companies. However, this may not be reassuring for P2P users, since any blocklist users (government or commercial) could be collaborating with media companies.

1.2 Measuring Spatial Properties of P2P Streams

Here, at a high level, we attempt to quantify the interesting characteristics of P2P traffic, especially regarding their spatial behavior, and to identify the challenges associated in measuring this kind of traffic.

We perform a study about the current measurement methodologies used to estimate the activity and size of P2P networks. We compile information in succinct manner which highlights the state-of-the-art in this regard. This is imperative to understand what we know about P2P networks and what we can expect to know about them. These two facets could possibly provide an insightful look into how to interpret results obtained by the measurement approaches which we have analyzed. Furthermore, we have also analyzed where are the peers located in the AS structure of the Internet when using these networks. We find that 92 to 98% of P2P flows end at tier 1 and tier 4 ASes. Furhter tier 1 and tier 4 ASes also provide maximum transit to P2P flows. Based on our findings we have provided a novel metrics with which P2P flows can be distinguished from other kinds of content streams such as Internet radio and http flows. Our mechanism can be used as a lightweight approach to make this distinction.

1.3 Analyzing Podcast Based Content Distribution

In this third part of the thesis, we analyze the characteristics of podcast content streams and develop a podcast traffic generation model for simulation purposes. Podcasts, a relatively new content distribution mechanism has become wildly popular since its introduction in 2004 [93, 94, 95] and is expected to garner an audience of nearly 56 million subscribers by 2010 [92], [86], [87]. One of the reasons for this rapid rise in popularity is that ”anyone” can publish a podcast. One does not need a content distribution infrastructure in place to serve content to large numbers of subscribers. Add to this free and ubiquitous web services which allow users to publish podcasts free of cost, and one can begin to understand the rationale for this atomic rise in popularity. In spite of this massive popularity quantifying the characteristics of podcast data streams has received little to no attention. In this part of our work, we measure podcast streams, analyze them and develop a traffic generator, SimPod, for simulation purposes. We find that Podcast file sizes are an order of magnitude greater than http file sizes.

1.4 An Overview of This Thesis

This thesis is logically divided into several chapters. Here we present a chapter-by-chapter breakdown of the text.

Chapter 2 Related Work:This chapter discusses the related work, the limitations and strengths of different approaches.

Chapter 3 Privacy Concerns in Peer-to-Peer (P2P) networks:This chapter discusses in details the issues related to privacy in P2P networks. We also describe the network setup and data collection process. Initially the chapter focusses on what level of monitoring users can expect ton encounter. Then, we analyze characteristics of monitoring nodes through analysis of real-world traffic traces.

Chapter 4 Measuring Spatial Properties of P2P content streams:This chapter discusses in details the spatial characteristics of P2P content flows. In this regard, we first discuss the various approaches to measuring P2P networks for the purposes of determining their scope. We highlight the limitations of each of the mechanisms. The, we move on to comparing characteristics of P2P flows with Internet Radio flows and http flows and propose a novel IR metric which can be used to distinguish P2P traffic from these other kinds of content flows.

Chapter 5 Podcast based content distribution and P2P:This chapter critically analyzes the important characteristics of podcast content flows. This new and emerging application is set to garner immense audiences in the near future. We dissect podcast flows from many dimensions, spatial, temporal and others. Finally, we present our results in the form of a simple to use traffic generator, SimPod. SimPod can integrate itself with standard topology generators to provide researchers with a way to simulate podcasts.

Chapter 6 Conclusion and Future Work

This chapter summarizes the contributions made in the thesis. We discuss possible implications of different assumptions made in the thesis. We discuss various future avenues into which this thesis can lead to.

Chapter 2

Related Work

This chapter describes the background and current work. Section 3 discusses relevant literature on privacy issues in P2P networks. Section 4 discusses the recent work on detecting where are P2P users located in the Internet and detection strategies for P2P flows. Section 5 discusses the recent work on content distribution via new and emerging applications.

2.1 Privacy Concerns in Peer-to-Peer (P2P) networks

P2P networks are a prevalent application in the Internet. There exists a plethora of P2P networks, such as FastTrack, Gnutella [19], BitTorrent, eMule/eDonkey along with extremely an long list of clients, written in all possible languages for nearly all operating systems, [18], [20]. P2P networks have recently been touted as the future of content distribution technologies [21]. However, the fact remains that, these overlay networks, still do act as significant enablers in the dissemination of copyrighted material. Organizations such as the

RIAA and MPAA have been extremely vociferous in their support for anti-P2P policies, since it is these organizations that lose out on revenue due to the exchange of copyrighted songs and movies [10], [12].

Recently, a slew of reports in the electronic and printed media have led to members of P2P communities pondering over the ramifications of such illegal resource sharing [23]. To reduce the threat of a possible lawsuit, users have resorted to downloading and deploying anti-detection software. This software blocks computers owned by these organizations from communicating with P2P users [13], [4]. This kind of software no longer allows entities monitoring P2P users to log the IPs of users. There is a large number of such free software, easily available, from popular websites, for many different P2P clients, networks and Operating Systems.

Previous work on modelling and analysis of P2P systems [29], [30], [31], has focused on developing a viewpoint based on performance metrics of such overlay systems. Our work differs greatly from these earlier efforts. Our goal is to quantify the probability of a P2P user of being tracked by entities listed on the most popular blocklists. To the best of our knowledge, our research is the first which specifically targets an in-depth study of whether such a threat is a reality for a generic P2P user. Moreover, our work is qunatifies *who do we talk to* while connected on these P2P networks, when sharing copyright-material. Additionally, we intend to verify reports suggesting that some so-called organizations enlisted by the RIAA *target UPs in preference to leaf nodes* [15], [16], in order to break the backbone of the entire overlay structure.

2.2 Measuring Spatial Properties of P2P content streams

P2P networks and their behavior have been the focus of active research efforts over the recent past. Efforts have been made to try and fathom the models being used by popular networks, [5], such as Gnutella, Edonkey [8] and BT [1], [3]. Studies carried out on such P2P networks, as highlighted in [4],[5],[6], [7] provide an in-depth perspective on how to discriminate traffic emanating due to P2P networks versus other Internet traffic. These methodologies range from payload identification, which involves filtering traces for particular hex strings, known beforehand, in the payloads of the packets captured. Other mechanisms employ parameters such as TCP flow holding time, average downloaded data size and others, to home in on possible P2P flows [7]. Research work regarding AS-AS interactions and P2P traffic have concentrated on interactions between a pair of ASs, while we attempt to develop a birds eye view mapping of where P2P users are located in the AS hierarchy. Furthermore, we compare P2P traffic with http traffic and Internet radio traces and highlight the differences between them. We employ custom designed tools interfaced with ethereal [2] in order to extract the AS information for each P2P flow.

2.3 Podcast based content distribution and P2P

Podcasting is rapidly gaining large audiences [85]. Individuals with access to the Internet are able to publish and distribute podcasts without the need for resource-rich infrastructure. This is a significant deviation from prevalent commercial organizations which provide multimedia content using a subscription model, or employ high speed servers and fat-bandwidth links to disseminate content to end-users [86], [87], [88].

Most content is audio but can be video as well, in the form of news feeds, interview transcripts, entertainment and radio shows. End-users subscribe to these feeds and RSS 2.0 enabled browsers and podcast aggregators automatically download files published by podcasters [95], [92].

One important research effort in this area describes a dynamic polling mechanism to reduce overhead incurred as a result of clients continuously polling content servers [116]. Our work differs significantly from this effort. We do not simulate end-user clients or propose a polling protocol. We focus on podcasts as a content delivery mechanism and quantify data and flow characteristics. Podcast data displays different characteristics when compared to content delivered by more traditional methods. Podcast data displays a different range for file sizes distributed to end-users compared to web/HTTP data. Research estimates report average page sizes for web pages to range from 60 to 605 KB [99], [100], [101]. This range is significantly different from podcast file size ranges by nearly an order of magnitude. Moreover, HTTP content displays a heavy tail Pareto distribution [102], different from podcast workload. Also, per-hour podcast traffic as observed from a client point of view follows a β distribution, unlike trends described for generic traffic in [118].

Also, real-time video and audio streaming is different from podcasting in terms of when data is transferred to end-users. Podcasting allows data to be disseminated, only when the content is published and hence data flows are bound by temporal characteristics of when content is published. This is not the case with other forms of content dissemination which are dependent solely on when the user wants to access the content. Research conducted in [**?**] verifies that user requests coming in for access to web objects follow a Zipf like distribution.

Characterization of Autonomous Systems (ASs) based on their degree-based ranks has been described in [106] and we employ these methods in our research. Efforts as caching performance and workload characterization of document data [102], segment based caching, with blockwise variable sized segments [103], caching based on data migration protocols, and event-driven paradigms [104] and summary cache [105] mechanisms could all be used to improve content delivery for podcasts.

Statistical background: We now define some statistical distributions [115], which will be used in subsequent sections.

1. β **distribution**: Formally defined as:

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

and

$$f(x;k,\theta) = x^{k-1}\frac{e^{\frac{-x}{\theta}}}{\theta^k\Gamma(k)}$$

where α and β both must be greater than zero, B defines the *Beta* function. This distribution is extensively employed in Bayesian statistics and is heavily used for PERT/critical-path-method based modeling. and k is the *shape parameter* and θ is the *scale parameter*, both greater than zero.

2. γ **distribution**: Formally defined by:

$$f(x;k,\theta) = x^{k-1}\frac{e^{\frac{-x}{\theta}}}{\theta^k\Gamma(k)}$$

where k is the *shape parameter* and θ is the *scale parameter*, both greater than zero.

3. **Bimodal distribution**: A bimodal distribution is a distribution with two different peaks, with two distinct values that measurements tend to center around. Such distributions have been used to model population dynamics for groups of individuals. For unimodal data, we observe:

$f(x_1) < f(x_2) < .... < f(x_m) > f(x_{m+1}) > f(x_{m+2}) > ... > f(x_N)$

While a bimodal distribution is apposite for cases which conform to the following criterion:

$f(x_1) < f(x_2) < .... < f(x_m) > f(x_{m+1}) > .. > f(x_{m+k}) < f(x_{m+k+1}) < .. < f(x_n) > f(x_n + 1) > ...f(x_N)$

A simple bimodal Gaussian distribution based on a threshold probability P, can be generated by the following algorithm:

Algorithm 1 BimodalGaussian x

```
$A \Leftarrow Vector[1 : X]$
$P \Leftarrow ThresholdProb$;
**for** $c = 1$ to $X$ **do**
    $Random = SampleUniform(1, 0, 1)$;
    **if** $Random > P$ **then**
        $A_c = SampleGaussian(\mu_1, \sigma_1)$;
    **else**
        $A_c = SampleGaussian(\mu_2, \sigma_2)$;
    **end if**
**end for**
```

Unimodal Pareto distributions do not seem to capture the file size distribution for podcast content data, while bimodal Gaussian distributions are capable of doing so, as will be seen subsequently.

Chapter 3

Privacy Concerns in P2P Networks

3.1 Introduction

Content providers, such as the RIAA and MPAA, have escalated their fight against illegal P2P sharing [7], [18], [19], [20],[26], [27] with the use of fear: there have been a number of lawsuits against individual P2P users [8], [9], [10], [11]. To make this more effective, these organizations and their collaborators have started "trawling" in P2P networks: creating "fake users", which participate in the network, and thus, identify users who contribute towards illegal content sharing. However, the extent of this deployment tactic has not been quantified up to now, and this forms the focus of our work.

In response to this approach, the P2P community has spawned several projects which attempt to identify such "fake users", and enable P2P users to avoid them. In more detail, there is a community based effort to maintain lists of suspect IPs, which are called *blocklists*.

Blocklists are published by organizations which provide anti-RIAA software or by groups which focus on security [14]. Additionally, a number of free, open-source, software projects have enabled P2P users to avoid these blocklisted IPs automatically. Such software is easy to download and is compatible with most popular P2P clients using various networks as BitTorrent, eDonkey-eMule, gnutella [2] [4], [13],[14] [36], [22], [32]. Note that it is not our intention here to examine how accurate and comprehensive these lists are, though this would be interesting and challenging future work. Our claim is that, the information that we use in our work is what is readily available to P2P users. We present Fig. 3.1.(a)-(b) which denote the significant numbers of P2P users who download and employ these blocklists to avoid contact themselves with fake users.

The question we attempt to answer is, how prevalent is the phenomenon of fake users. Simply put, how likely is it that a user without running any additional software will run into such a "fake user"? The answer to this question can help us: (a) understand the effort that content providers are putting in trawling P2P networks, and (b) justify the effort of the P2P community to isolate "fake users". To the best of our knowledge, this phenomenon has not been quantified before.

We conduct an extensive measurement study, employing PlanetLab [17] for a period of 90 days. We analyze more than 100GB of TCP header data, monitoring clients connected to the Gnutella network and use the most popular blocklists on the Internet [4], [14], [36]. We create and deploy P2P clients which insert about 100 popular song queries from well-known music charts [42], [35], [34] into the P2P network. Hereonwards, we refer to IPs of fake users listed on these blocklists as blocklisted IPs and users exchanging data with them as *being tracked*. A blocklisted IP is said to be *hit* every time a user interacts with it. Our results can be summarized as follows:

1. **Pessimistic result:** A user without any knowledge of blocklists, will almost certainly be tracked by blocklisted IPs. We found that **all** our clients exchanged files with blocklisted IPs. In fact, of all distinct IPs contacted by any client, 12-17% were found to be listed on blocklists.

2. **Optimistic results:** We find that *a little information goes a long way*: Avoiding just the top 5 blocklisted IPs reduces the chance of being tracked to about 1%. This is a consequence of a skewed preference distribution: we find that the top 5 blocklist ranges encountered during our experiments contribute to nearly 94% of all blocklist encounters.

3. **Most blocklisted IPs belong to government or corporate organizations:** We quantify the percentage of hits, to blocklisted IPs of each type, i.e. government and corporate, educational, spyware proliferators and Internet advertisement firms. We find that the number of hits which belong to government and corporate lists are nearly 2.5 times more than educational, spyware and adware lists.

4. **Very few blocklisted IPs belong directly to content providers:** We find that 0.5% of all blocklisted IP hits could actually be traced back to media companies, such as Time Warner Inc.

5. **Geographical bias:** We find that there is a geographical bias associated with how users encounter entities listed on blocklists. Users located on the two opposite coasts, east and west, of mainland US, in Europe and Asia, hit blocklisted entities according to different paterns.

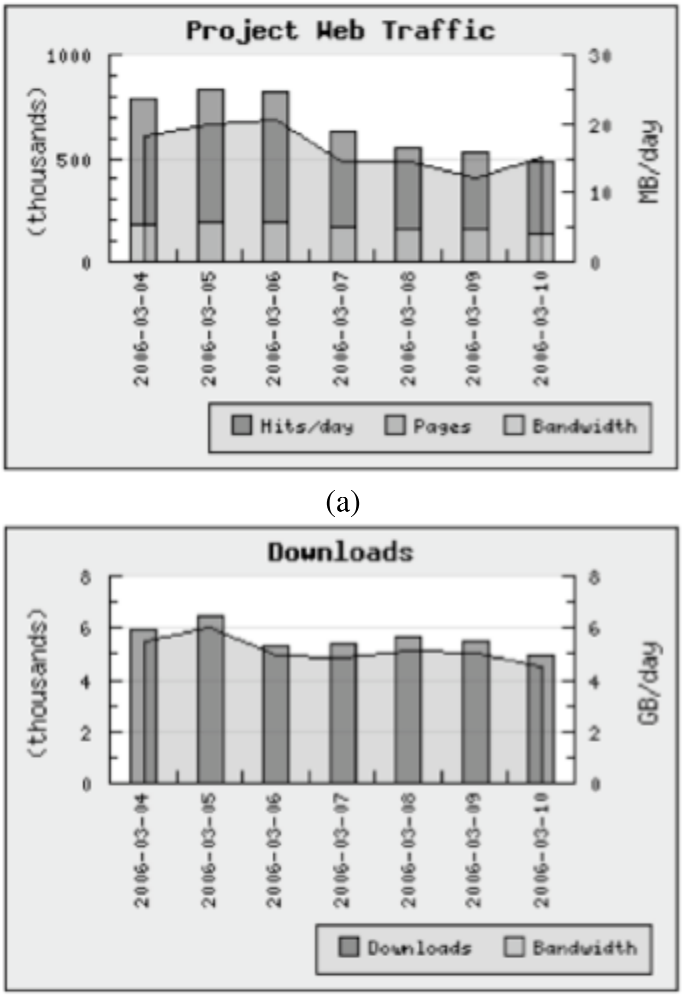
6. **Equal opportunity trawling:** We find that Ultra-peers [1] and leaf nodes have equal probability of associating with a blocklisted IP, with less than 5% variation in the average number of distinct blocklisted IP hits. This comes in contrast to the popular belief that UPs are tracked more aggressively by blocklisted entities [15], [16], than leaf users.

The rest of this work is organized as follows. Section II details the relevant literature applicable to our research, followed by Section III, which discusses the experimental setup. This is followed up by section IV, which investigates the probability of a user being tracked. Section V deals with unearthing geographical bias followed by Section VI which addresses the Ultra/Super peer versus leaf node debate. This is followed by the conclusion.

3.2 Who is Watching?

In this section, we discuss the experimental setup we employ and quantify the most prevalent on blocklisted entities in P2P networks. We find that:

[1] Ultra-peers are high bandwidth nodes that act as local centers, facilitate low bandwidth *leaf* nodes, and enable the scalability of gnutella-like networks.



(a)

(b)

Figure 3.1: Activity statistics for Peerguardian, compiled 10th March 2006:(a) The total webtraffic directed towards the Peerguardian webpage at Sourceforge. (b) The amount of downloads for the software, ranging from 4.5 to 6.5 GB (approx.) per day, from the same site. [4]

1. The majority of the most active blocklisted entities encountered are hosted by organizations which want to remain anonymous.

2. Content providers such as the RIAA do not participate in large scale eavesdropping into P2P networks using their own IPs.

We initiate our experiments in a manner so as to be able to emulate the typical user and yet be able to measure large scale distributed network-wide inter-node interaction characteristics of such P2P networks. We measure statistics based on trace logs compiled from connections initiated using PlanetLab to gather traces in a geographically distributed environment. The duration of measurements spanned more than 90 days, beginning January 2006. We initiate connections using 50 nodes spread not only across the continental US (35 nodes), but also Europe (10 nodes), and Asia (5 nodes) in order to determine any geographical nuances associated with, which entities on blocklists seems to be more active than others, in specific locations. We customized mutella 0.4.5 clients [33], a vanilla console based gnutella client, and intitiate connections to the Gnutella network. Moreover, clients were made to switch intechangeably from UP to leaf modes in order to verify if network wide inter-node behavior of UPs is significantly different from leaf nodes.

Our queries in the P2P network were based on lists of popular songs, from Billboards hot 100 hits [34], top European 50 hits [35] and Asian hits [42]. Each node injected about 100 queries during every run. In the process, we analyzed more than 100GB of TCP header traces, using custom scripts and filters to extract relevant information which helps us develop a deeper insight into who do we interact with, while sharing resources on P2P networks.

Before we present results obtained from our measurements we must discuss what BOGON IPs [40] mean as they hold special siginificance to the collected information. BOGON is the name used to describe IP blocks not allocated by IANA and RIRs to ISPs and organizations plus all other IP blocks that are reserved for private or special use by RFCs. As these IP blocks are not allocated or specially reserved, such IP blocks should not be routable and used on the internet, however some of these IP blocks do appear on the net primarily used by those individuals andorganizations that are often specifically trying to avoid being identified and are often involved in such activities as DoS attacks, email abuse, hacking and other security problems.

Table I lists the top fifteen entities that we encounter on the P2P network while exchanging resources, throughout the duration of our active trace collection. Surprisingly, these entities operate from BOGON IP ranges. This observation is made on the basis of the various popular blocklist resources, and suggests that *these sources deliberately wish to conceal their identities while serving files on P2P networks*, by using up IP ranges which cannot be tracked down using an IP-WHOIS lookup to locate the operator employing these anonymous blocks. Only three out of the top fifteen entries in table I do not use unallocated BOGON IP blocks and are listed on PG lists [4], the rest of the BOGON entities, are listed on both Trustyfiles [36] and Bluetack [14] lists. Most of the BOGON IP ranges point to either ARIN or RIPE IP ranges. We must however mention that these BOGON IP ranges were found to point back to these generic network address distribution entities at the time of our experiments. It is quite possible that these ranges may have now been allocated to firms or individuals and may no longer remain truly anonymous. We observe that 99.5% of blocklisted IPs contacted either belong to BOGON, commercial entities, educational institutions while only about 0.5% of all blocklisted IPs we came in contact with could actually be traced back to record companies, such as Time Warner Inc. This is an indication of the small presence of record companies themselves, snooping on P2P users in a proactive manner.

FUZION COLO listed in Table I, is understood to propagate self installing malware, and in general as an anti P2P entity [37], [38], [41]. xeex [39] is more of a mystery. It hosts an inconspicious site which provides absolutely no information as to what the company is really involved in. Going by the discussion groups hosted on the PG website, xeex does turn up frequently in blocklist hits for a large number of users. Other individuals or organizations deliberately employing BOGON IPs to participate in the exchange of resources on P2P networks are certainly attempting to hide, possibly from the RIAA. Another vein of reasoning would suggest that they could be the ones who keep track of what users download.

$Rank$	$Top15HitRanges$	$Type$
1	72.48.128.0-72.235.255.255	Bogon
2	87.0.0.0-87.31.255.255	Bogon
3	88.0.0.0-88.191.255.255	Bogon
4	72.35.224.0-72.35.239.255	FuzionColo
5	71.138.0.0-71.207.255.255	Bogon
6	70.229.0.0-70.239.255.255	Bogon
7	70.159.0.0-70.167.255.255	Bogon
8	70.118.192.0-70.127.255.255	Bogon
9	216.152.240.0-216.152.255.255	xeex
10	216.151.128.0-216.151.159.255	xeex
11	70.130.0.0-70.143.255.255	Bogon
12	87.88.0.0-87.127.255.255	Bogon
13	71.66.0.0-71.79.255.255	Bogon
14	87.160.0.0-87.255.255.255	Bogon
15	70.82.0.0-70.83.255.255	Bogon

Table I: Listing of top 15 blocklist entities encountered on P2P network.

Table II and Table III display the top five entities on the (a) educational and research institutions list and the (b) government and commercial organizations lists. We observe that FuzionColo and xeex appear prominently in this categorization along with two other commercial organizations which host servers on ed2k and gnutella networks. We find that hits to entities listed on commercial and government blocklists are much more frequent than hits on any other different kind of blocklists such as Internet ad companies, educational institutions and others. Even though the number of IPs which belong explicitly to content providers such as the RIAA may be small, the fact that IPs listed on commercial and government blocklists are providing content to P2P users is of concern. The scenario in which commercial organizations are hired by content providers to collect user profile data cannot be ruled out. Furthermore, the possibility that these commercial organizations, such as the ones listed in Table III are not aware of P2P traffic emanating from their servers does not seem very plausible since some of these blocklisted entities kept tracking our clients nearly every time files were exchanged. It is clear that these commercial IP ranges, which serve files to P2P users, have a very large cache of popular in-demand media and have extremely low downtime. In fact, the number of hits to commercial and government blocklisted entities is nearly 2.5 times greater than hits to any other kind of blocklisted IPs.

Rank	*Top 5 Educational Hit Ranges*
1	152.2.0.0-152.2.255.255-Univ. of N. Carolina
2	64.247.64.0-64.247.127.255-Ohio University
3	129.93.0.0-129.93.255.255-Univ. of Nebraska
4	128.61.0.0-128.62.255.255-Georgia Tech
5	219.242.0.0-219.243.255.255-CERNET

Table II: Listing of top 5 educational entities encountered on P2P networks

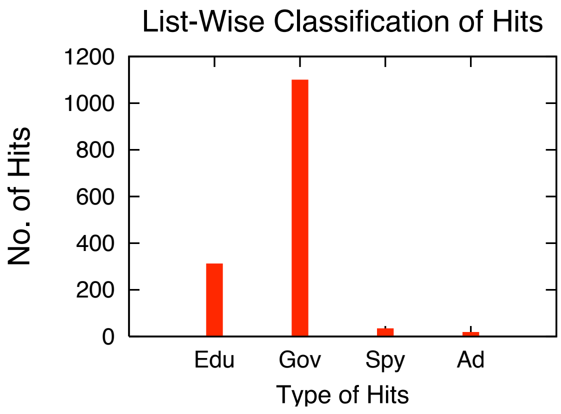


Figure 3.2: Classification of blocklist hits according to their type. We observe that hits on the commercial and government blocklist is significantly larger than hits on the other blocklists.

Rank	*Top 5 Commercial Hit Ranges*
1	72.35.224.0-72.35.239.255-FuzionColo
2	216.152.240.0-216.152.255.255-xeex
3	216.151.128.0-216.151.159.255-xeex
4	38.113.0.0-38.113.255.255-Perf.SystemsInted2k
5	66.172.60.0-66.172.60.255-Netsentryed2kserver

Table III: Listing of top 5 commercial entities encountered on P2P networks

3.3 Probability of being tracked

In this section, we estimate the probability of a typical P2P user being tracked by entities listed on these blocklists. This gives an idea of what percentage of entities encountered while surfing P2P networks are not considered trustworthy. We observe the following during our study:

1. 100% of all our nodes were tracked by entities on blocklists and on average, 12-17% of all distinct IPs contacted by any of our clients were listed on blocklists.

2. Popularity of blocklisted IPs tracking P2P users follows an extremely skewed distribution.

As illustrated in Fig. 3.3, the percentage of IPs listed on blocklists is quite significant, about 12-17% of all distinct IPs contacted, per node. In fact, this trend was reflected throughout the duration of our measurements, which suggests that the presence of blocklisted entities on P2P networks is not an ephemeral phenomenon. Furthermore, we observe that the frequency of popularity for blocklisted entities follows a skewed distribution as displayed in Fig. 3.4a. A small number of entities register a large number of hits while most blocklisted entities are infrequently visible on P2P networks. This fact is of great consequence to users who wish to avoid contact with blocklisted entities and thus reduce their chances of running into anti-P2P entities. *Avoiding the five most popular blocklisted IPs leads to a drastic reduction in the number of hits to blocklisted IPs, approximately by 94%.* This interesting statistic is displayed in Fig. 3.4b. In fact, **avoiding just these top 5 blocklisted IPs can reduce the chances of a user being tracked significantly, down to nearly 1%**. Users can use this fact to tweak their IP filters to increase their chances of safely surfing P2P networks and bypassing the most prevalent blocklisted entities. This is critical considering that, a naive user, without any information of blocklists will almost certainly be tracked by blocklisted entities. Also, the fact that 100% of all nodes regardless of geographical location were tracked by

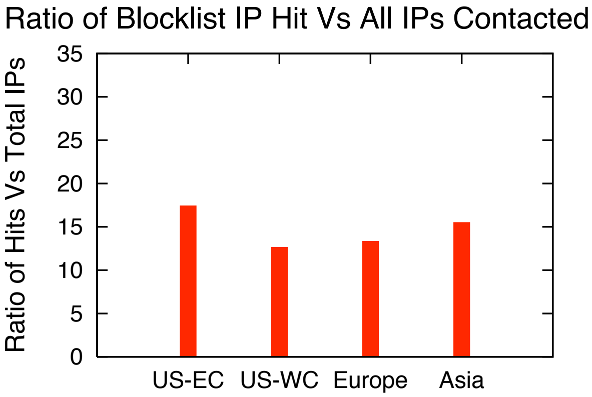


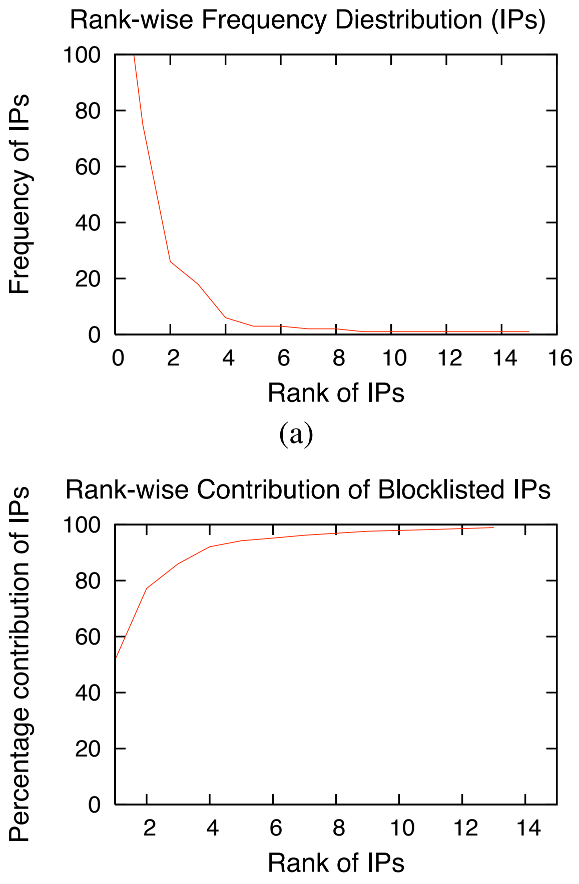
Figure 3.3: Percentage of distinct blocklist IPs contacted out of the total number of distinct IPs logged.

blocklisted IPs, indirectly points to the completeness of the blocklists we collected from the most popular sources.

3.4 Geographical Distribution

In this section, we focus attention towards the issue regarding whether geographical bias exists in our active measurements with respect to entries on blocklists being encountered while our clients connect to the P2P networks from various geographical locations. To achieve this, we needed a geographically diverse set of P2P users. We employed over 50 different nodes on PlanetLab, encompassing the continental US, Europe and Asia. We monitor individually, PlanetLab nodes located in the continental US as nodes situated on the east coast (US-EC) and on the west coast (US-WC), to observe if there is any variation in behavior within mainland US and, surprisingly, we do observe such a difference as discussed below.

In Fig. 3.5a, we study the effect of geographical location on how blocklisted IPs track P2P



(a)

(b)

Figure 3.4: (a)Frequency of popularity of blocklisted IPs, following a skewed distribution.(b)Percentage contribution by Blocklisted IPs. The 5 most popular blocklisted IPs contribute to nearly 94.2% of all blocklist hits.

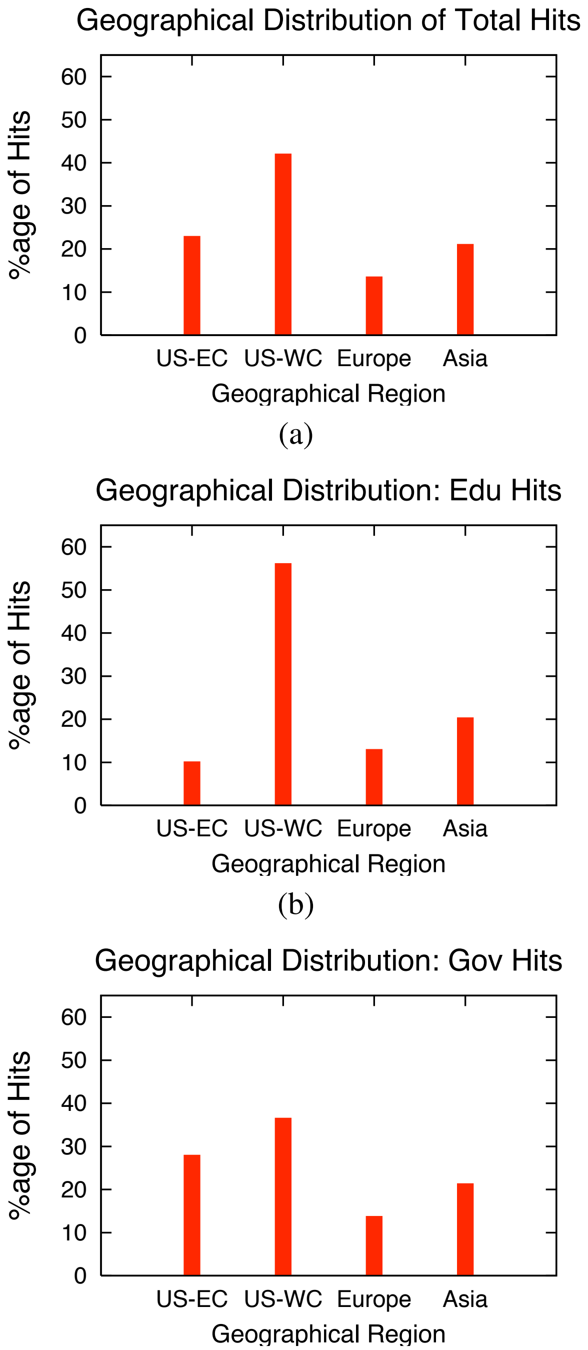
users. We observe that the percentage of blocklisted IP hits is highest in US-WC followed by US-EC, Asia and Europe. *The percentage of blocklisted IP hits, per node, as a percentage of total hits to IPs contacted by each node, located on the US west coast seems to be nearly twice of that for nodes located on US east coast*. This suggests that users accessing the P2P network from these two vantage points, within the mainland US, encounter different levels of tracking activity. We believe this observed inequality stems from the difference in user behavior and possible difference in levels of monitoring activities by entities on the blocklists could directly be responsible for such a skewed trend. Fig. 3.5b depicts the distribution of blocklisted IP hits from the ”educational” range, comprising of academic and research institutions. Again, we observe a similar trend, nodes located on US-WC have a up a higher percentage of blocklist hits compared to nodes located on US-EC, Asia and Europe. In fact, the difference in measurements between US-WC and US-EC is more than five times than that of readings gathered from US-EC.

Fig. 3.5c depicts the distribution of blocklisted IP hits in the government and commercial domain. Once again, we observe that the probabilities for nodes situated on US-WC are higher than nodes on US-EC, Asia and Europe. The period of observation, the UTC time when data was logged, the number of queries in the P2P network, the order in which queries were injected were identical for all nodes. This suggests that, throughout the duration of our experiments, *a consistent skewed distribution between US west coast and US east coast can be due to difference in user behavior and the differing degree of local tracking activity in these different geographical settings*. Nodes located in Europe consistently registered a lower number of blocklisted IP hits when compared to nodes located in Asia. We always attempt to maintain a balance while logging data using PlanetLab and deploy our code on nearly the same number of nodes in different geographical settings, log data during synchronized time periods using automated scripts bootstrapped via crontab. The only difference while gathering measurements in these settings was that we used different lists of queries which were injected into the P2P network for nodes located in separate continents. For nodes located in Europe, we constructed query lists based on European 50 hits [35], and for nodes in Asia we constructed query lists based on Asian hits [42]. The magnitude of difference observed between nodes in Europe and Asia was found to be more or less consistent across the different types of blocklisted IPs. However, they were significantly different from measurements gathered across the mainland US.

3.5 Effect of role on the probability of being tracked

This section delves into whether, according to popular perception in P2P communities [15], [16], the probability of being tracked by blocklisted entities varies with the role played by a P2P node. The question we answer is: *are UPs are tracked with higher probability by entities on blocklists versus regular leaf nodes*. We find that **the role of the node does not seem to have an effect on its probability of being tracked by blocklisted IPs**. To examine this, we repeatedly configured nodes to shift from UP to leaf mode and back over a number of cycles in order to obtain connectivity patterns for each mode of operation. The uptime for each mode was identical and experiments were repeated to smoothen out any temporal fluctuations in observed data. We observe in Fig. 3.6a the mean number of distinct IPs contacted by leaf nodes and UPs in variousgeographical locations. We find that leaf nodes, located in the US, seem to interact with a larger number of distinct IPs than do UPs. However, this is not the case in either Europe or Asia, where UPs connect to larger number of distinct IPs than leaves. This observation could be due to the false perception, hyped primarily in the US that UPs are being watched with more vigor by entities on the blocklists compared to leaf users. Since significant legal action against users of P2P networks has been directed towards users in the US, it is obvious that peers would refrain from voluntarily switching their P2P client's mode of operation to become a UP. Therefore, we see a much lesser intensity of UP interaction within P2P networks in the US. While in Asia, where the threat of legal action has yet to create a dent in the minds of P2P users, it is evident that users will hardly shy away from switching clients to UP mode or at least deliberately prevent clients from doing so. Hence, we observe much larger numbers of peers communicating with UPs in Asia. We believe that the same vein of thought holds true for the scenario for Europe based nodes, albeit to a lesser extent.

Ultra-Peers do not encounter more blocklisted entities than Leaf-nodes in a consistent manner. In Fig. 3.6b, we compare the percentage of blocklisted IP hits as recorded by UPs and leaf nodes. The percentage shows how many of the total number of IPs encountered are blocklisted IPs. This metric depicts if there is any correlation between UPs being tracked preferentially over leaf nodes irrespective of geographical location. We find that UPs in US-WC encounter higher numbers of blocklisted IP ranges versus leaf nodes. This trend is consistent with Europe based nodes. However for US-EC and Asia based nodes we observe that UPs encounter lesser numbers of blocklist IPs compared to leaf nodes. In fact we observe less than 5% variation in the average number of blocklisted IP hits by UPs versus leaf nodes on these P2P networks and thereby do not find any supporting evidence for claims of UPs being preferentially tracked by entities on these lists vis-a-vis leaf nodes. From our experiments we understand that a UP has nearly the same probability of running into blocklisted entities as leaf users and do not find any significant variation in the number of blocklisted entities contacted by either. It must be noted though that our experiments do suggest a difference in P2P user behavior between US-WC and US-EC as has been discussed in previous sections.

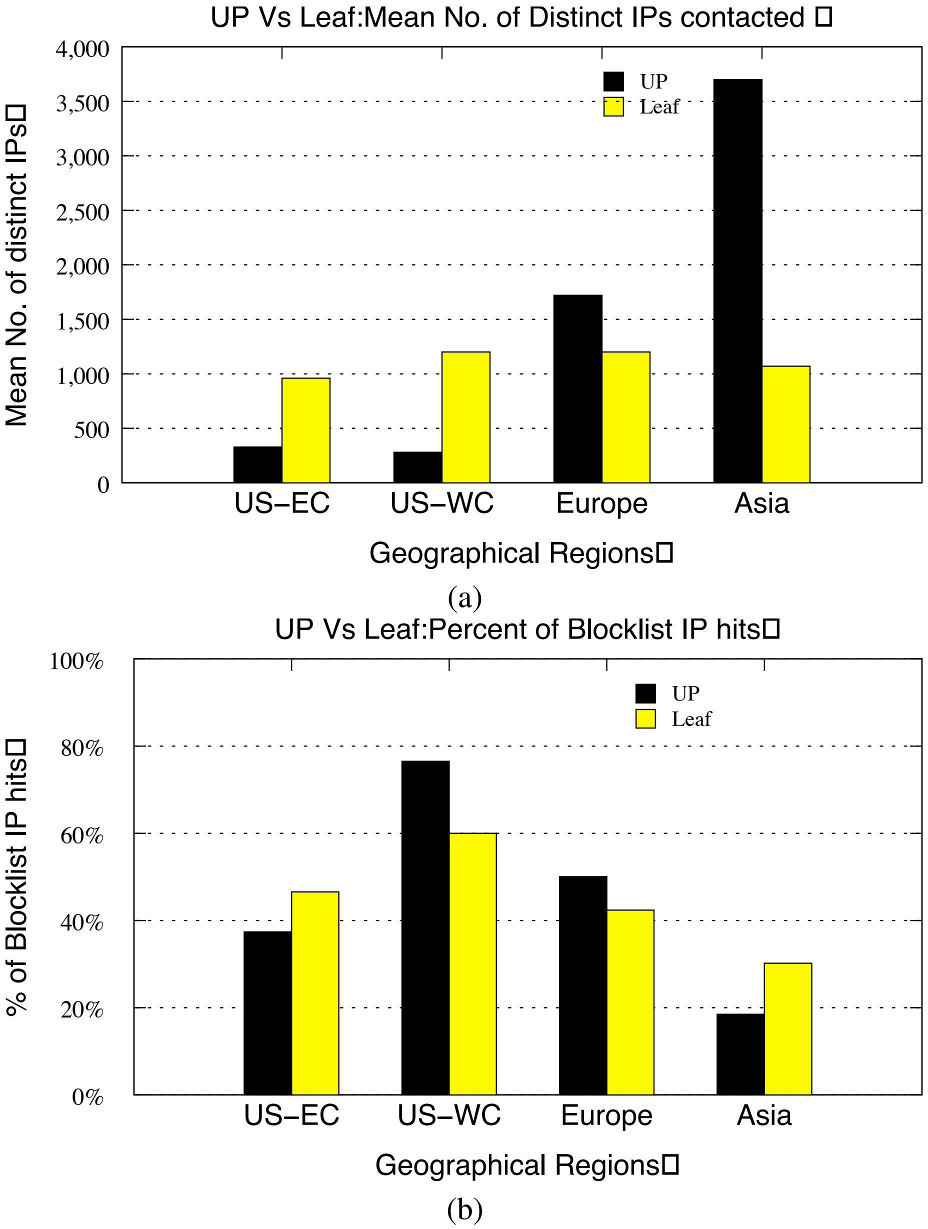


(a)

(b)

(c)

Figure 3.5: (a)Distribution of Blocklisted IPs contacted in different geographical zones, (b)Distribution of blocklisted IPs contacted, on Educational lists, in different geographical zones, (c)Distribution of blocklisted IPs contacted, on Government and Commercial lists, in different geographical zones.



(a)

(b)

Figure 3.6: UP Vs Leaf:The black bar signifies UPs while the yellow bar signifies leaf users:(a) Comparison of average number of distinct IPs contacted by UPs and leaves, (b) Comparison of percentages of blocklisted IP hits as encountered by UPs and leaf users.

Chapter 4

Measuring Spatial Properties of P2P content streams

4.1 Analyzing P2P-network measurement approaches

4.1.1 Introduction

The eDonkey protocol has been a dominant protocol in the P2P arena since its appearance in 2000. Its popularity appears to have peaked in 2004 when it overtook Fasttrack as the most popular P2P protocol. During 2005, it was reported that eDonkey was serving approximately two to three million users who were sharing 500 million to two billion files via 100 to 200 servers [43]. The popularity and the server-based operation were bound to create problems for eDonkey: in February 21 of 2006, and June 16 of 2007, some of its major servers were raided and shut-down by the authorities. However, eDonkey has not been eliminated. The question we attempt to answer in this report is what is the current state of eDonkey now.

Measuring eDonkey is fairly challenging as it has a hybrid architecture: a server-based operation but with multiple servers. At the same time, the servers have been known to communicate with each other, and in some variations of the protocol client (e.g. eMule), the clients can have some autonomy in their operation and not require the constant presence of a server. Further, it is hard to measure the "actual" number of available files and peers which are willing to share them because of the high percentage of free-riding phenomenon, presence of users who do not share and only download off the network, in eDonkey [50]. Several studies and efforts have attempted to estimate the current level of activity in eDonkey [52, 51, 49, 55, 50].

The goal of this report is to contribute to the state of the art of what we know and what is possible to know about eDonkey. The report spans three areas: (a) what is currently known, (b) what are the limitations of the current measurement methods, and (c) what more can we learn and how. In more detail, we address the above three issues as follows.

a. Current Knowledge: We assess what the community currently knows about eDonkey. We find that their is a discrepancy in the number of users reported by various efforts. The number of clients can range from 230,000 [49] to 9 million users [55]. However, the trend of clients using this network is clearly increasing.

The number of distinct files in this network can be as high as 11 million [50] and most peers share only about 10 files [51] or less. A popular media file can be spread over as many as 45,000 clients. Interestingly, the number of free riders, can be very significant, nearly 84%

of the total population [50]. This behavior poses a serious challenge for reliably determining the number of peers in the network who are uploading or downloading files. Furthermore, the injection of a few replicated decoys can lead to significant perturbations in the network and can be used as a successful poisoning strategy [52].

The conclusion here is that eDonkey continues to represent a significant force in the P2P world, and it should be considered in any study for P2P traffic.

b. Evaluating the measurement approaches. We categorize the different approaches to measuring eDonkey and evaluate their capabilities, accuracy, and limitations. First, we distinguish the methods into **active** and **passive** techniques because both these classes answer questions from different perspectives.

Active methods attempt to crawl the network by identifying as many servers as they can, and then querying the servers with many different files in order to identify the size and the number of files [51, 53]. Clearly, the identification of all the servers and the immense scale of querying a few hundred million files are the key limitations of these methods [51].

Passive methods start with the data on an Internet link and attempt to identify the percentage of eDonkey traffic. A key limitation of this method is that identifying P2P traffic is far from trivial [62, **?**]. Identifying which traffic is eDonkey within the P2P traffic is even harder, and very few efforts attempt to do this.

Overall, we find that current measurement studies seem to face significant limitations, which is shown from the wide variations of their estimates of the eDonkey activity.

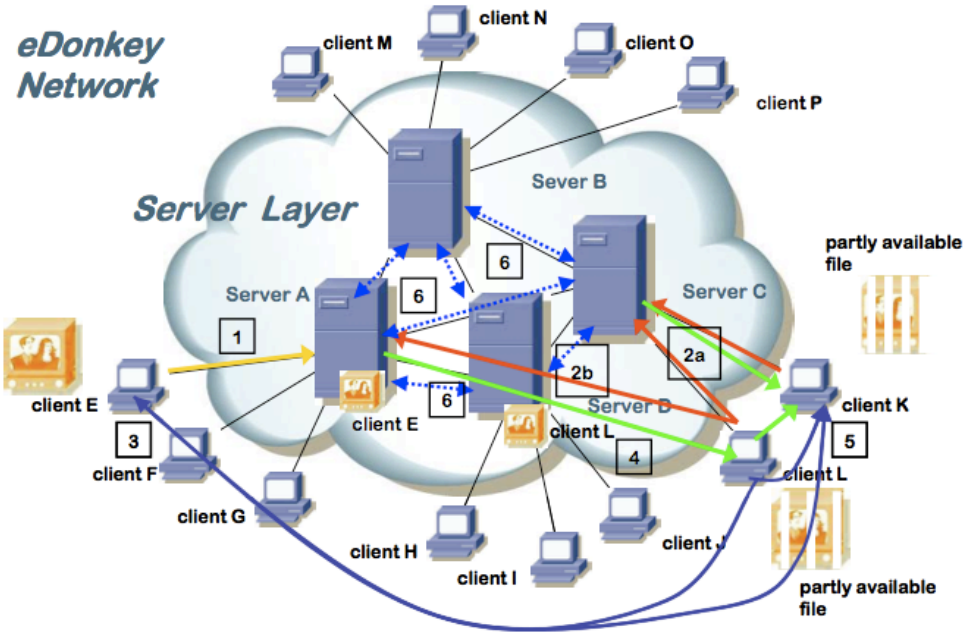


Figure 4.1: eDonkey Architecture from [47]: Servers interact with each other, clients connect to servers to find other clients.

4.1.2 The eDonkey Architecture

In this section we provide a high level overview of the eDonkey network followed by a more granular analysis of each component of the network.

General Architecture

The eDonkey network comprises of servers and clients. Each server provides a lookup service for files in the network on behalf of clients. The eDonkey network does not rely on a single central server. It is a distributed architecture wherein multiple different servers can help clients locate the files they need. In this *hybrid* P2P architecture clients can obtain parts of a file from different peers at the same time and can also upload parts of a file, which it is currently downloading in parallel. It should be noted that no files are transmitted by servers, they simply act as index directories for locating resources in the P2P network. Furthermore, every eDonkey client can opt to be a server.

File Searching

In eDonkey, a set of dedicated servers allow peers to search for files. Upon startup, a peer connects to a server, which then assigns a unique ID to that peer. There are two types of IDs assigned by each server, HighID and LowID. A peer on an open host, not behind a NAT, is assigned a HighID, which is the decimal representation of the hosts IP address. A peer on a guarded host, behind a NAT, is assigned a LowID, which is an arbitrary 32-bit unique number managed by its connected server. Each server determines whether a host should be assigned a HighID or LowID based on the result of a proprietary probing. Comparing the number of HighID peers and LowID peers gives a good estimate on the population of eDonkey peers running on guarded hosts. When a peer X queries its connected server for a file, the server returns a list of peers, along with their IDs, that have the requested file. Upon receiving a set of matched peers from the server, peer X starts to download the file from these peers in parallel.

In Fig. 4.1 we present an example of how the eDonkey network operates. When a client connects to the eDonkey service, it logs on to one of the servers (using a TCP connection) and registers all files it is willing to share, represented by the arrow labeled 1 in Fig. 4.1. Each server keeps a list of all files shared by the clients connected to it. When a client searches for a file, represented by arrow labeled 2a, it sends the query to its main server. The server returns a list of matching files and their locations. The client may resubmit the query to another server, represented by the arrow labeled 2b, if none or an insufficient number of matches have been returned. The major communication between client and server is typically implemented by TCP connections on port 4661. Additional communication between clients and servers, e.g. further queries and their results, are transmitted via UDP on port 4665.

File Downloading

Files on the eDonkey network are uniquely identified using MD4 root hash of an MD4 hash list of the file. This treats files with identical content but different names as the same, and files with different contents but same name as different.

Files are divided in full chunks of 9,728,000 bytes (9500 x 1024 bytes) plus a remainder chunk, and a separate 128-bit MD4 checksum is computed for each. That way, a transmission error is detected and corrupts only a chunk instead of the whole file. Furthermore, valid downloaded chunks are available for sharing before the rest of the file is downloaded, speeding up the distribution of large files throughout the network. A file's identification checksum is computed by concatenating the chunks' MD4 checksums in order and hashing the result. In cryptographic terms, the list of MD4 checksums is a hash list, and the file identification checksum is the root hash, also called top hash or master hash. It is possible for two different chunks or files to have the same checksum and thus appear the same, but the chance of that happening is so small that for all practical purposes it never happens, and checksums are considered unique identifiers.

When an eDonkey client decides to download a file, it first gathers a list of all potential file providers and then asks the providing peers for an upload slot, see arrow with label 3 in Fig. 4.1. Upon reception of a download request, the providing client places the request in its upload queue. A download request is served as soon as it obtains an upload slot. eDonkey clients may restrict their total upload bandwidth to a given limit. An upload slot comes available when a minimum fair share of the upload limit is possible. When an upload slot is available, the providing client initiates a TCP connection to the requesting client, negotiates which chunk of the file is exchanged, and transmits the data. As mentioned earlier, the eDonkey protocols splits the file into separate pieces, denoted as chunks. The consuming client can reassemble the file using the chunks or parts of chunks. A client can share a file as soon as it has received a complete chunk, see arrow with label 4 in Fig. 4.1. A major feature of eDonkey is that the consuming client may operate in the multiple source download mode, see arrow marked 5 in Fig. 4.1. In this mode, the downloading client issues in parallel two or more requests to different providing clients and retrieves data in parallel from the providers.

Since an eDonkey client may leave the eDonkey service at any time, the requesting client has to renew its download request periodically otherwise the requests are dropped. In order to reliably check the availability of a client, the eDonkey protocol uses TCP connections on port 4662 for the communication between the clients. A client-to-client connection is terminated by the eDonkey application after an idle period of 40sec. It is worth to be mentioned here, that other P2P file sharing applications like Bearshare or KaZaA have implemented similar multiple source download schemes.

Inter-server Message Exchange

The main eD2k server is the Lugdunum eserver [44]. The Lugdunum server software was created by reverse engineering edonkey protocol and redesigned from scratch.

The communication between eDonkey servers is very limited, see arrow marked 6 in Fig. 4.1. The servers contact each other periodically but with small frequency in order to announce themselves and to send back a list of other servers. In this way the servers maintain an updated list of working servers and affirm the search efficiency of the eDonkey service.

Additional Functionality

eMule, an open source version of the eDonkey client, includes a pure P2P client *source-exchange* capability, allowing a client with a High ID (i. e., with incoming eD2k connections not blocked by a firewall) to continue downloading (and uploading) files with a high number of sources for days, even after complete disconnection from the original Kad or eD2k servers that handled the original requests. This source-exchange capability is designed to reduce the load on servers by two thirds or more for files that have a large number of seeds, or sources (other clients) for the files. The original eDonkey client by MetaMachine does not support source exchanges [51, 44].

Currently, it seems that the eDonkey network cannot function without the presence of servers to perform a lookup service. However, the eMule client which embodies the source-exchange capability is able to do so. In fact, the eMule client publishes file information on two separate DHT based networks, eD2K and KAD thereby increasing the visibility and availability of a file to its peers.

4.1.3 Measurements

In this section we list the possible measurements which may be carried out on the eDonkey network. We group the methods into three important types of measurements. The observations here are a synthesis of many studies. In the appendix, we provide a concise and informative summary of each research study, the methodology they employ and the take-home points of the study. note that the tables are a critical evaluation of each study.

Active Measurements

We list the kind of metrics that these methods can measure and provide a detailed overview of what we know for that metric.

- **Number of Users**: This can be measured by using an eDonkey client to continuously query various servers for the IDs of peers which host parts of a file. This method is very popular in practice for estimating the population of peers interested in a particular file [52, 56, 50, 54, 47].

  What do we know: The estimates for network size of the eDonkey network vary significantly. Recent reports estimate the size of the eDonkey network to be about 2.5-

3 million users [50, **?**] and 9 million users [55]. Previous studies have reported smaller numbers of clients, 230,000 in 2003 [49]. However, *the increasing trend of users* has been highlighted in [54]. Further it has been reported that 25-36% of eDonkey clients are situated behind NATs [51]. This makes it very difficult to accurately estimate the size of the network.

Possible Limitations: This method has some limitations when estimating the size of the network in its entirety. The first complication arises from which files should be chosen to probe the network and for how long and how many servers should be probed. Only if all files on the network were queried to all the servers we could be confident about measuring the size of the network

- **Number of Servers**: This can be measured by keeping track of the various lists which publish information about available servers on the network. This information can be supplemented by monitoring responses exchanged between servers when they transfer server lists among each other.

  What do we know: The number of servers reported varies from 35-50 [51] to about 250 [**?**].

- **Number of downloadable files:** This can be measured by querying the network with the help of an eDonkey client for different files. The various hashes for the files which are returned provide an idea of how many distinct copies of a particular file are present in the network at any time. Furthermore, this could help to understand the "variety" of files in the P2P network. A first step in this direction has been described in [52] and [51].

  What do we know: The number of distinct files on the eDonkey network has been reported to be as large as 11 million [50]. In [51], it has been reported that popular avi media files can be hosted on as many as 45,000 peers in the eDonkey network. The median sizes of files is about 64 MB and most peers share less than 10 files. It has been clearly mentioned in this study that the methodology followed for uncovering the number of files hosted by individual peers is not very accurate. Additionally it has also been reported that the number of free-riders, peers which only download and do not share, can be as high as 84% of the total population [50]. This directly reduces the number of actual downloadable files in the network.

- **Content Pollution**: Pollution (a.k.a. content-poisoning or interdiction) is the distribution of fake files in order to "annoy" users of P2P networks and push them towards acquiring content via legitimate methods.The number of versions of a file which are bogus can provide insight into the quality of content in the network. The goal here is to measure how many "bad" versions of a file are distributed and what is the probability that a generic user will be able to download a good copy.

What do we know: Pollution schemes for eDonkey have not been studied extensively. It has been reported that pollution schemes in eDonkey can be effective if a small number of copies is carefully disseminated in the network [52].

Possible Limitations: Due to the prevalence of file poisoning practices in P2P networks, and the absence of a content rating mechanism in eDonkey, it is difficult to determine if a file is fake or not without actually downloading it [52]. This implies that one would have to manually verify if a file is good or bad. This requires significant manpower for a measurement study and so far there has been no large scale measurement studies.

Passive Measurements

- **Profiling eDonkey users**: Logging the activity of a client or a server can be done by using a packet sniffer at the machine where the client or the server are deployed. Monitoring an eDonkey server allows the recording of information regarding all the query requests send to the server, as well as the set of files that each user is willing to share. To the best of our knowledge, no efforts have been made to profile eDonkey users.

Possible Limitations: Identifying eDonkey traffic is challenging [62]. Additionally, profiling clients always has to deal with the issue of "representativeness". It is always difficult to generalize results from a few observations.

- **Monitoring the network core**: Passively monitoring (sniffing) from a large (backbone or access-link) link of an ISP. Such data sets require a way to detect eDonkey traffic[1]. To address this problem, we can use tools such as the BLINC P2P classifier [62] combined with a payload-based classifier. Such a payload-based classifier is available with the BLINC tool. Further, a new highly-effective tool from UCR, Graption [**?**], detects P2P flows by forming traffic dispersion graphs between clients and servers. In Table 4.1, we present an example of how effective Graption is. Initial results show that it is reasonably effective in detecting eDonkey flows from a backbone trace.

  Possibilities and advantages: By monitoring a highly aggregated link, such as a backbone link, we can extract detailed information regarding the relative popularity of eDonkey compared to other P2P protocols.

  Also, analyzing backbone traces can allow the extraction of additional statistics regarding the flow level behavior of eDonkey. Such statistics can include: total flow duration, packet size distributions, bandwidth usage for each flow, use of both TCP and UDP

[1] Even thought currently there are many methods for the detection of P2P applications based on flow statistics (e.g., packet sizes) or host behavioral profiling, these methods are limited to the labeling of flows as P2P or non-P2P and currently do not attempt to map a flow to a particular P2P application of origin. In order to achieve this we have to use: (a) payload inspection for the detection of eDonkey client-to-server or server-to-client headers; or (b) an exhaustive list of eDonkey server IPs, and label all the flows towards and from those IPs.

Name	*% in Flows*	*% in Bytes*	*% in Packets*
Gnutella	0.95(6.78)	0.17(1.59)	0.81(6.32)
eDonkey	2.96(21.16)	2.22(21.17)	2.84(22.21)
FastTrack	0.55(3.92)	0.74(7.10)	0.97(7.61)
Soribada	7.76(55.44)	0.07(0.63)	0.97(7.63)
MP2P	0.41(2.93)	0.01(0.14)	0.07(0.53)
BitTorrent	0.60(4.26)	4.59(43.81)	4.37(34.24)
All P2P	13.85	9.19	12.10

Table 4.1: Application Breakdown: Values in parenthesis show the percentage of each P2P application over the entire P2P identified traffic. The trace is collected from an OC-48 link from the Palo Alto Internet eXchange Center (PAIX) in 2004.

protocols by peers, etc.

Possible Limitations: The main limitations are (a) it is extremely challenging to obtain non-anonymized IP backbone traces and packet payload information from ISPs due to privacy concerns and (b) to accurately separate eDonkey traffic from other kinds of P2P traffic.

	Time	Ref.	Page Ref.	Technique
1	Aug 03	[47]	49	Passive: Measurement based approach. The Internet connection of the university is a 155Mbps link to the German Research Network (DFN).
2	Nov 03	[63]	50	Passive: Signature based approach. Two traces were collected. First trace was on an access network to a major backbone and contains typical Internet traffic.The total traffic volume was 128 GB of compressed data and corresponded to 4.58 million TCP connections. Second was a VPN Trace, on a T3 (45 Mbps) link which has a low probability of carrying P2P traffic containing 1.8 Terabytes of data in 2.8 billion packets.
3	Nov 03	[49]	51	Active: Crawling the eDonkey network to get the peer's cache contents information. DE FR ES US IT NL IL GB TW PL CH Others.
4	Dec 03 – Feb 04	[50]	52	Active: Crawling based approach. Europe.
5	Feb 04	[51]	54	Modifying the existing eMule client to build a modified client which uses passive and active probes on the network. North America, Europe, and Asia.
6	Dec 04	[52]	55	Active measurements of the network. 50 nodes located in 18 different countries in North and South America, Europe, Asia, and Oceania.
7	Apr 06	[53]	56	Active: Crawling the network. Europe, China, United States.
8	Nov 06	[54]	57	Passive: Payload and Port based Analysis (for eDonkey ports analysed are: 4661, 4662, 4665,4672, 5662, 40662, 14662). 3,090 users connected to France Telecom backbone network carrying ADSL traffic; both to and from the backbone and ADSL area. Analysis done only for TCP traffic.
9	Mar 03 – Mar 06	[55]	58	Passive: Protocol Signature Analysis. Europe.
10	Nov 03	[56]	59	Active probing. Europe.

Table 4.2: Summary of eDonkey Measurement Techniques.

Table 4.3: Summary of Research on eDonkey Measurement.

Published as	By Whom	Results	When	Where	Techniques
Lecture Notes in Computer Science, ACM May 11, 2004	A Measurement-based Traffic Profile of the eDonkey File-sharing Service Kurt Tutschku, Institute of Computer Science, University of Wurzburg, Germany. tutschku@informatik.uni-wuerzburg.de	-In total almost 3.5 million flows were investigated carrying 295 Gbyte of data (non-download and download). Only 2.24% of all connections were download connections, but carrying 705% of the total traffic. - The average observed eDonkey flow size was 86Kbytes, with the average size of download streams (2.48Mbytes) two orders of magnitudes larger than the average size of non-download streams (16.7Kbytes).It doesn't change much for inbound and outbound flows. - Download flow size decreases strongly because they are limited due to the segmentation of files into chunks and due to the application of the multiple source download principle. Non download flows don't matter much as they are typical signalling flows to renew requests. - eDonkey flows, however, showed that a large number of connections have a significant idle period before the TCP connection is terminated. The eDonkey traffic doesn't seem to worsen the ”mice and elephants”phenomenon. - A majority of the observed traffic is locally and not world-wide distributed. A large number of established connections, however, do not necessarily mean a high traffic volume. This feature is caused by the eDonkey protocol requirement to renew download requests.	Aug-2003	- These experiments were conducted over a duration of 296h on a 100Mbps, half duplex FE link connecting the department with the university's campus LAN. - The Internet connection of the university is a 155Mbps link to the German Research Network (DFN).	Passive: Measurement based approach

Continued on next page

Table 4.3 – continued from previous page

Published as	By Whom	Results	When	Where	Techniques
Proceedings of the 13th international conference on World Wide Web table of contents New York, NY, USA Date: 2004	Accurate, Scalable InNetwork Identification of P2P Traffic Using Application Signatures Subhabrata Sen, Oliver Spatscheck, Dongmei Wang	- eDonkey mostly operates through its default ports as the study found 2149.84 MB of eDonkey traffic on default ports using their signatures (102%), whereas its presence on non-default ports was only 4.10% (much less than other prevalent techniques).	Nov-2003	Two traces were collected. First trace was on an access network to a major backbone and contains typical Internet traffic.The total traffic volume was 128 GB of compressed data and corresponded to 4.58 million TCP connections. Second was a VPN Trace, on a T3 (45 Mbps) link which has a low probability of carrying P2P traffic containing 1.8 Terabytes of data in 2.8 billion packets.	Passive: Signature based approach

Continued on next page

Table 4.3 – continued from previous page

Published as	By Whom	Results	When	Where	Techniques
In: 3rd International Workshop on Peer-to-Peer Systems, 2004 also in Book Series Lecture Notes in Computer Science, Jan 2005	Clustering in Peer-to-Peer File Sharing Workloads France Switzerland Cambridge, UK F. Le Fessant, S. Handurukande, A.-M. Kermarrec and L. Massoulie	- 230,000 eDonkey clients were discovered. 55,000 of them were connected during a 3 days period. 37,000 clients could clearly be identified as distinct clients, among which 25,000 clients shared no files at all. A total of 923,000 different files were studied. - On file popularity: a few files are extremely replicated while a large number is not replicated at all. - Multimedia files, audio and video dominate the proportion of files in the system: Audio files represent the largest number of files (48% against 16% for video), while in terms of size video files are dominant (67% against 16% for audio). - Other measures show that MP3 files dominate in number, while AVI files largely dominate in terms of size. - Geographical clustering : peers requesting a given video file may in a large proportion of cases download it from peers in their own country, thus achieving low latency and network usage compared to downloading it from a randomly chosen peer. They found for 60% of the files, more than 80% of the replicas are located in the main country. - Performance of P2P search for audio files can be greatly improved if we exploit interest-based locality: if two peers share interests, (in other words if the contents of their cache overlap significantly) the search mechanism can be significantly improved by having these peers connect to one another and first send their requests to each other.	Nov-2003	DE FR ES US IT NL IL GB TW PL CH Others	Active: Crawling the eDonkey network to get the peer's cache contents information

Continued on next page

Table 4.3 – continued from previous page

Published as	By Whom	Results	When	Where	Techniques
Proceedings of the 2006 EuroSys conference	Peer Sharing Behaviour in the eDonkey Network, and Implications for the Design of Server-less File Sharing Systems S. B. Handurukande!, A.-M. Kermarrec, F. Le Fessant, L. Massoulie and S. Patarin	- They found 2.5 million connections to peers and browsed their cache contents. The total volume of shared data thus discovered (with multiple counting) was over 350 terabytes. - It is stated that then current eDonkey servers could handle more than 200,000 connected users (provided they have the necessary bandwidth). - They identified over one million distinct peers (29% in France, 28% in Germany, 16% in Spain and only 5% in the US), and 11 million of distinct files. Also the average clients share 5 new files per day. - A large majority of eDonkey users are in Europe, with France and Denmark heading the list. - Specialization of the eDonkey network is in downloading large files. It is found that 40% of the files are less than 1MB, 50% are between 1 and 10 MB, the typical range forMP3 files. Only 10% of the files are larger than 10MB. However, among files with popularity larger than 5 (respectively, larger than 10), about 45% (respectively, 55%) of the files are larger than 600MB (these are typically DIVX movies), less than 20% are between 1MB and 10MB, the typical range for MP3 files, and less than 20% have sizes between 10 and 600MB, the typical range for complete MP3 albums, small videos and programs. - The number of files and the amount of data shared per client, with and without free-riders: We observe that free-riding (approximatively 80% of the clients) is very common in eDonkey. Most of the remaining clients share a few files, 80% of the non free-riders share less than 100 files, but these are large files, since less than 10% of non free-riders share less than 1GB. This feature is common to most p2p file sharing systems, but this phenomenon is even more pronounced in the eDonkey network, again reflecting its specialization towards large files.	Dec-03 to Feb-04	Europe	Active: Crawling based Approach

Continued on next page

Table 4.3 – continued from previous page

Published as	By Whom	Results	When	Where	Techniques
		- The ranks of popular files tend to remain stable over time, even though the degree of popularity measured by the number of replicas may decrease. - The sharing distribution is skewed: a few clients share most of the files, both in terms of size and number of files. - Geographical clustering patterns: 50% of files with an average popularity greater than or equal to 20 have all their sources in the same country, while this is the case for only 10% of the files with a popularity greater than or equal to 50. In both figures, there is a clear distinction between popular and non popular files. The geographical clustering tends to be more pronounced for non popular files. - The workload is highly dynamic: clients share a roughly constant number of files over time, but the turnover is high. - The presence of clustering, both geographical and semantic, between clients in peer-to-peer file sharing workloads; with semantic clustering is even stronger for rare files. In semantic clustering if some clients have a small number of files in common, the probability that they will share another one is very high. For audio files, it is observed in particular that unpopular files are more subject to clustering.			

Continued on next page

Table 4.3 – continued from previous page

Published as	**By Whom**	**Results**	**When**	**Where**	**Techniques**
In Proceedings of Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE Publication Date: 29 Nov.-3 Dec. 2004	Characterizing Guarded Hosts in Peer-to-Peer File Sharing Systems, University of Michigan, Ann Arbor, USA {wenjiew, hschang, azeitoun, jamin}@eecs.umich.edu.	- The overall average of guarded host percentage on the experimental servers is about 36%. The ratio of guarded hosts on each server varies significantly from 0% to 81%. - About 25-36% of eDonkey users are located on guarded hosts and share a nontrivial amount of files on peer-to-peer systems. On guarded hosts the availability of each file decreases roughly by one-third. - The prevalence of guarded hosts not only inhibits fair sharing among peers, but also may interfere with efficient overlay construction for overlay-based peer-to-peer systems.	Feb-2004	North America, Europe, and Asia	Modifying the existing eMule client to build a modified client which uses passive and active probes on the network.

Continued on next page

Table 4.3 – continued from previous page

Published as	By Whom	Results	When	Where	Techniques
Proceedings of the 6th ACM conference on Electronic commerce Vancouver, BC, Canada Publication: 2005	Content Availability, Pollution and Poisoning in File Sharing Peer to Peer Networks christin@sims.berkeley.edu, andreas@weigend.com chuang@sims.berkeley.edu	- There are significantly more returns in eDonkey and it produces results extremely quickly: after two minutes, for nearly all queries, the sender has received over 85% of all query returns. After 3.5 minutes, the network has returned virtually all responses to every query. This is because of the small response time by the highly centralized topology in eDonkey. - eDonkey has very high temporal stability i.e measure of how the users perception of the available content changes over time. After 24 hours, there is a 50% chance that a given user perceives a specific movie file as still being present on the network. - Spatial stability i.e the probability that an item be seen at n hosts, decreases linearly. This is because different servers in eDonkey provide significantly different returns. Indeed, the very small number of servers in eDonkey translates into a high probability that several of our hosts are connected to the same server. - Perceived content replication, that is, the number of copies of a given file that are found in a search initiated by a node, generally follows a power law distribution. Users have a quite accurate perception of the relative availability of different files. - Regarding download completion time i.e the total time needed to successfully complete a download, the eDonkey clients initially lag behind other popular P2P clients (like FastTrack, Gnutella) before catching up. This is due to the credit system used in the downloading algorithm in eDonkey i.e peers which upload more traffic get more credits, and can in turn download files from a larger number of peers. Such a credit system mildly penalizes newcomers, and thus the results. - In eDonkey poisoning techniques that leave a clear statistical signature should be relatively easy to detect and combat.	Dec-2004	50 nodes located in 18 different countries in North and South America, Europe, Asia, and Oceania.	Active measurements of the network

Continued on next page

Table 4.3 – continued from previous page

Published as	By Whom	Results	When	Where	Techniques
Fifth International Conference on Grid and Cooperative Computing Workshops, 2006. GCCW06	Crawling the eDonkey Network Peking University, Beijing, P.R.China {yangj, mah, songweijia, cj, zclfly}@pku.edu.cn	- Information collected for over 1 million clients with unique IP addresses and almost 5 million unique files, it was found that only a small number of clients made their shared files available to their client, meaning eDonkey client softwares usually disable the view shared files functionality by default, and users are reluctant to let other users view their files. - 758 servers were discovered with majority lying in DE, US and NL. - 1, 287, 082 clients with unique IP addresses and 6, 184, 635 files were discovered. The number of unique files is 4, 937, 087 with total size of 4.67 1014 bytes. Most of these files are audio and video files	April-2006	Europe, China, United States	Active: Crawling the network

Continued on next page

Table 4.3 – continued from previous page

Published as	**By Whom**	**Results**	**When**	**Where**	**Techniques**
International Teletraffic Congress 2007: Ottawa, Canada; Also in Book Series Lecture Notes in Computer Science September 04, 2007	Measurement Based Modeling of eDonkey Peer-to-Peer File Sharing System {Fabrice.Guillemin, Walid.Saddi}@orange-ftgroup.com	- eDonkey is still the dominating p2p protocol. - eDonkey peers massively (about 60%) use standard port numbers (port numbers: 4661,4662,5662,40662,14662), even if large data transfers can take place with dynamic port numbers i.e about 60% of elephants are on standard port numbers. - The amount of traffic transferred on conventional and assimilated ports represents 56% (resp. 66%) of whole eDonkey traffic in the downstream (resp. upstream) direction. - eDonkey gives rise to a huge amount of signaling activity (observed by tracking mice), and actual data transfers take place only on a small number of TCP connections - eDonkey is characterized by a high activity (in terms of packets and not in terms of volume) on standard or assimilated port numbers, certainly due to signaling. This experimental observation may be used for p2p identification. Even if p2p protocols have been designed with many features to hide p2p transactions, many real customers run the most basic versions of protocols. - Due to the sharing principles implemented by the eDonkey protocol, congestion limits the expansion of the network (formation of a core of congested nodes) and, thus, asymptotically slows down the dissemination of an object, even if the object is eventually shared among all peers.	Nov-2006	3,090 users connected to France Telecom backbone network carrying ADSL traffic; both to and from the backbone and ADSL area. Analysis done only for TCP traffic.	Passive: Payload and Port based Analysis (for eDonkey ports analysed are: 4661, 4662, 4665,4672, 5662, 40662, 14662)

Continued on next page

Table 4.3 – continued from previous page

Published as	By Whom	Results	When	Where	Techniques
International Conference on Internet Surveillance and Protection, 2006.	Public Domain P2P File-sharing Networks Measurements and Modeling Polytechnic University of Valencia (Spain), Jaime Lloret, Juan R. Diaz, Jose M. Jimenez and Fernando Boronat	- Measurements taken in 2006, show that eDonkey network can have variations of 9 million of users in one hour. - The total amount of data shared inside P2P networks is not dependent with the number of files shared inside the network. - During the experimental period number of files in an eDonkey network has exponentially increased to more than 1billion (the fastest among all P2P networks analyzed). - Also the number of users in eDonkey network increased exponentially to more than 20 million (fastest among the P2P networks analysed). - The average files per user through the years remained almost constant at 100, falling much less in the comparative study with other P2P networks. - All type of files (big files like videos, DVD images) are shared with high correlation between the number of users. More users connected are for 1 - 20 hours. - In the eDonkey is the network with most total amount of shared data inside because many of them are big size files like videos, DVD images, and so on. The total number of users connecting to the P2P filesharing networks is growing. Therefore, the number of users increasing Internet traffic, due to the use of these networks, is growing. - eDonkey in March 2006 had no less than 52 files per user.	March-03 to March-06	Europe	Passive: Protocol Signature Analysis

Continued on next page

Table 4.3 – continued from previous page

Published as	By Whom	Results	When	Where	Techniques
Proceedings of the 11th workshop on ACM SIGOPS European workshop.	Exploiting semantic clustering in the eDonkey P2P network, S. Handurukande, A.-M. Kermarrec, F. Le Fessant and L. Massoulie.	- For semantically related neighbours, the hit rate is very encouraging (with just 5 peers detected hit rate is close to 30%). - The hit rate when neighbours are selected randomly is very low and remains low even with a significant number of peers. - The hit ratio is influenced by the presence of very generous uploaders who contribute large amount of files, and strongly indicates that the semantic clustering truly exists. - Transitivity of the semantic relationship: Results do not exhibit better results for 2nd level semantic search in particular, if the important factor is the number of peers contacted. However, the fact that 2nd-level semantic search provides similar results as when the same number of semantic neighbours peers are contacted during the first hop, shows that semantic links tend to automatically cluster semantically-related peers. - The hit ratio is negatively impacted when maintaining a peer profile (audio, video etc) per type of file.	Nov-2003	Europe	Active probing

Table 4.4: Additional Details for Research on EDonkey Measurement.

By Whom	Results	Where	What	How
A Measurement-based Traffic Profile of the eDonkey File-sharing Service Kurt Tutschku, Institute of Computer Science, University of Wurzburg, Germany. tutschku@informatik.uni-wuerzburg.de	- The average observed eDonkey flow size was 86Kbytes, with the average size of download streams (2.48Mbytes) two orders of magnitudes larger than the average size of non-download streams (16.7Kbytes).It doesn't change much for inbound and outbound flows. - Download flow size decreases strongly because they are limited due to the segmentation of files into chunks and due to the application of the multiple source download principle. Non download flows don't matter much as they are typical signalling flows to renew requests. - eDonkey flows, however, showed that a large number of connections have a significant idle period before the TCP connection is terminated. The eDonkey traffic doesn't seem to worsen the "mice and elephants"phenomenon. - A majority of the observed traffic is locally and not world-wide distributed. A large number of established connections, however, do not necessarily mean a high traffic volume. This feature is caused by the eDonkey protocol requirement to renew download requests.	- This experiments were conducted over a duration of 296h on a 100Mbps, half duplex FE link connecting the department with the university's campus LAN. - The Internet connection of the university is a 155Mbps link to the German Research Network (DFN).	Provide a traffic profile for the eDonkey service with focus: 1. on the distinction of non-download traffic and download traffic; which are distinguished by eDonkey / eMule protocol opcodes 'OP_SENDINGPART' or 'OP_COMPRESSEDPART', 2. the "mice and elephants" characteristic in eDonkey, and 3. the origin and destination of eDonkey flows.	Measurements were performed on flow level using TCP dump which was configured to record all TCP flows on the eDonkey client-to-client port '4662' and the flows were later classified.

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Accurate, Scalable InNetwork Identification of P2P Traffic Using Application Signatures Subhabrata Sen, Oliver Spatscheck, Dongmei Wang	- eDonkey mostly operates through its default ports as the study found 2149.84 MB of eDonkey traffic on default ports using their signatures (102%), whereas its presence on non-default ports was only 4.10% (much less than other prevalent techniques).	Two traces were collected in Nov-03. First trace was on an access network to a major backbone and contains typical Internet traffic.The total traffic volume was 128 GB of compressed data and corresponded to 4.58 million TCP connections. Second was a VPN Trace, on a T3 (45 Mbps) link which has a low probability of carrying P2P traffic containing 1.8 Terabytes of data in 2.8 billion packets.	1. Provide an efficient approach for identifying the P2P application traffic through application level signatures. 2. To efficiently identify the application level signatures by examining some available documentations, and packet-level traces; and then utilize them to develop online filters that can efficiently and accurately track the P2P traffic even on high-speed network links.	They derive application layer signatures for P2P protocols; decompose the P2P signatures into fixed pattern matches at fixed offsets within a TCP payload and variable pattern matches with variable offset within a TCP payload.

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Clustering in Peer-to-Peer File Sharing Workloads France Switzerland Cambridge, UK F. Le Fessant, S. Handurukande, A.-M. Kermarrec and L. Massoulie	- 230,000 eDonkey clients were discovered. 55,000 of them were connected during a 3 days period. 37,000 clients could clearly be identified as distinct clients, among which 25,000 clients shared no files at all. A total of 923,000 different files were studied. - On file popularity: a few files are extremely replicated while a large number is not replicated at all. - Multimedia files, audio and video dominate the proportion of files in the system: Audio files represent the largest number of files (48% against 16% for video), while in terms of size video files are dominant (67% against 16% for audio). - Other measures show that MP3 files dominate in number, while AVI files largely dominate in terms of size. - Geographical clustering : peers requesting a given video file may in a large proportion of cases download it from peers in their own country, thus achieving low latency and network usage compared to downloading it from a randomly chosen peer. They found for 60% of the files, more than 80% of the replicas are located in the main country. - Performance of P2P search for audio files can be greatly improved if we exploit interest-based locality: if two peers share interests, (in other words if the contents of their cache overlap significantly) the search mechanism can be significantly improved by having these peers connect to one another and first send their requests to each other.	DE FR ES US IT NL IL GB TW PL CH Others	To study the clustering properties and provide a map of the content shared by the peers of eDonkey.	The crawler, implemented by modifying an existing eDonkey client, namely MLdonkey, runs two concurrent tasks: 1. discovering eDonkey clients (by connecting to as many eDonkey servers as possible requesting their list of clients) and 2. scanning their contents (by attempting to connect to every eDonkey client that is discovered. If it succeeds, it obtains the unique identifier of the client and requests its list of shared files).

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Peer Sharing Behaviour in the eDonkey Network, and Implications for the Design of Server-less File Sharing Systems S. B. Handurukande, A.-M. Kermarrec, F. Le Fessant, L. Massoulie and S. Patarin	- They found 2.5 million connections to peers and browsed their cache contents. The total volume of shared data thus discovered (with multiple counting) was over 350 terabytes. - It is stated that then current eDonkey servers could handle more than 200,000 connected users (provided they have the necessary bandwidth). - They identified over one million distinct peers (29% in France, 28% in Germany, 16% in Spain and only 5% in the US), and 11 million of distinct files. Also the average clients share 5 new files per day. - A large majority of eDonkey users are in Europe, with France and Denmark heading the list. - Specialization of the eDonkey network is in downloading large files. It is found that 40% of the files are less than 1MB, 50% are between 1 and 10 MB, the typical range for MP3 files. Only 10% of the files are larger than 10MB. However, among files with popularity larger than 5 (respectively, larger than 10), about 45% (respectively, 55%) of the files are larger than 600MB (these are typically DIVX movies), less than 20% are between 1MB and 10MB, the typical range for MP3 files, and less than 20% have sizes between 10 and 600MB, the typical range for complete MP3 albums, small videos and programs. - The number of files and the amount of data shared per client, with and without free-riders: We observe that free-riding (approximatively 80% of the clients) is very common in eDonkey. Most of the remaining clients share a few files, 80% of the non free-riders share less than 100 files, but these are large files, since less than 10% of non free-riders share less than 1GB. This feature is common to most p2p file sharing systems, but this phenomenon is even more pronounced in the eDonkey network, again reflecting its specialization towards large files.	Europe	A study of semantic and geographical clustering in the workload gathered by crawling the eDonkey network.	Actively probing the cache contents of peers distributed over several countries, mostly in Europe (DN, NL).

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
	- The ranks of popular files tend to remain stable over time, even though the degree of popularity measured by the number of replicas may decrease. - The sharing distribution is skewed: a few clients share most of the files, both in terms of size and number of files. - Geographical clustering patterns: 50% of files with an average popularity greater than or equal to 20 have all their sources in the same country, while this is the case for only 10% of the files with a popularity greater than or equal to 50. In both figures, there is a clear distinction between popular and non popular files. The geographical clustering tends to be more pronounced for non popular files. - The workload is highly dynamic: clients share a roughly constant number of files over time, but the turnover is high. - The presence of clustering, both geographical and semantic, between clients in peer-to-peer file sharing workloads; with semantic clustering is even stronger for rare files. In semantic clustering if some clients have a small number of files in common, the probability that they will share another one is very high. For audio files, it is observed in particular that unpopular files are more subject to clustering.			

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Characterizing Guarded Hosts in Peer-to-Peer File Sharing Systems, University of Michigan, Ann Arbor, USA {wenjiew, hschang, azeitoun, jamin}@eecs.umich.edu.	- The overall average of guarded host percentage on the experimental servers is about 36%. The ratio of guarded hosts on each server varies significantly from 0% to 81%. - About 25-36% of eDonkey users are located on guarded hosts and share a nontrivial amount of files on peer-to-peer systems. On guarded hosts the availability of each file decreases roughly by one-third. - The prevalence of guarded hosts not only inhibits fair sharing among peers, but also may interfere with efficient overlay construction for overlay-based peer-to-peer systems.	North America, Europe, and Asia	To measure the prevalence of guarded hosts in two popular peer to peer file sharing systems, eDonkey and Gnutella, and study the characteristics of their shared files.	1. Their modified eMule client connects to a server, and asks for files with well-known suffixes (avi, mp3, mpg, and wmv). 2. From the list of files returned for each suffix, the client requests the 15 highest-ranked files in terms of their availability. 3. Each request prompts the server to return a list of peers storing a requested file. 4. From these lists, they distinguish between HighID and LowID peers. 5. Their client stays connected to each server for 20 minutes to repeat this process three times and then randomly switches to another server to get a reliable host statistic on guarded host population.

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Content Availability, Pollution and Poisoning in File Sharing Peer to Peer Networks christin@sims.berkeley.edu, andreas@weigend.com chuang@sims.berkeley.edu	- There are significantly more returns in eDonkey and it produces results extremely quickly: after two minutes, for nearly all queries, the sender has received over 85% of all query returns. After 3.5 minutes, the network has returned virtually all responses to every query. This is because of the small response time by the highly centralized topology in eDonkey. - eDonkey has very high temporal stability i.e measure of how the users perception of the available content changes over time. After 24 hours, there is a 50% chance that a given user perceives a specific movie file as still being present on the network. - Spatial stability i.e the probability that an item be seen at n hosts, decreases linearly. This is because different servers in eDonkey provide significantly different returns. Indeed, the very small number of servers in eDonkey translates into a high probability that several of our hosts are connected to the same server. - Perceived content replication, that is, the number of copies of a given file that are found in a search initiated by a node, generally follows a power law distribution. Users have a quite accurate perception of the relative availability of different files. - Regarding download completion time i.e the total time needed to successfully complete a download, the eDonkey clients initially lag behind other popular P2P clients (like FastTrack, Gnutella) before catching up. This is due to the credit system used in the downloading algorithm in eDonkey i.e peers which upload more traffic get more credits, and can in turn download files from a larger number of peers. Such a credit system mildly penalizes newcomers, and thus the results. - In eDonkey poisoning techniques that leave a clear statistical signature should be relatively easy to detect and combat.	50 nodes located in 18 different countries in North and South America, Europe, Asia, and Oceania.	1. To know the factors that influence the sensitivity of a network to poisoning and pollution, and 2. measure the content availability in the P2P networks in the absence of observable poisoning, and 3. separately characterize the effects of different poisoning strategies on each network.	Different queries (songs, movies, software) were injected in each network at multiple times.

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Crawling the eDonkey Network Peking University, Beijing, P.R.China {yangj, mah, songweijia, cj, zclfly}@pku.edu.cn	- Information collected for over 1 million clients with unique IP addresses and almost 5 million unique files, it was found that only a small number of clients made their shared files available to their client, meaning eDonkey client softwares usually disable the view shared files functionality by default, and users are reluctant to let other users view their files. - 758 servers were discovered with majority lying in DE, US and NL. - 1, 287, 082 clients with unique IP addresses and 6, 184, 635 files were discovered. The number of unique files is 4, 937, 087 with total size of 4.67 1014 bytes. Most of these files are audio and video files	Europe, China, United States	A client-based approach that actively probes (part of) the eDonkey network.	Experimental eDonkey client consisting of: 1. server crawler (Crawler 1), 2. client and file crawler (Crawler 2), and 3. client resource crawler (Crawler 3).

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Measurement Based Modeling of eDonkey Peer-to-Peer File Sharing System {Fabrice.Guillemin, Walid.Saddi}@orange-ftgroup.com	- eDonkey is still the dominating p2p protocol. - eDonkey peers massively (about 60%) use standard port numbers (port numbers: 4661,4662,5662,40662,14662), even if large data transfers can take place with dynamic port numbers i.e about 60% of elephants are on standard port numbers. - The amount of traffic transferred on conventional and assimilated ports represents 56% (resp. 66%) of whole eDonkey traffic in the downstream (resp. upstream) direction. - eDonkey gives rise to a huge amount of signaling activity (observed by tracking mice), and actual data transfers take place only on a small number of TCP connections - eDonkey is characterized by a high activity (in terms of packets and not in terms of volume) on standard or assimilated port numbers, certainly due to signaling. This experimental observation may be used for p2p identification. Even if p2p protocols have been designed with many features to hide p2p transactions, many real customers run the most basic versions of protocols. - Due to the sharing principles implemented by the eDonkey protocol, congestion limits the expansion of the network (formation of a core of congested nodes) and, thus, asymptotically slows down the dissemination of an object, even if the object is eventually shared among all peers.	3,090 users connected to France Telecom backbone network carrying ADSL traffic; both to and from the backbone and ADSL area. Analysis done only for TCP traffic.	To know the structure of the eDonkey network, when the N peers have joined the system given there is a community of N peers sharing an object according to the principles of eDonkey.	1. A mathematical model is built to represent the behavior of an "idea" eDonkey (i.e getting rid of all technological constraints and considering the community in isolation) community sharing an object. 2. These restrictive assumptions nevertheless allow to point out two regimes of an eDonkey network: expansion and collapse. 3. The transition between these two modes depends upon a congestion factor, which is related to the size of objects and the uplink transmission capacities of peers.

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Public Domain P2P File-sharing Networks Measurements and Modeling Polytechnic University of Valencia (Spain), Jaime Lloret, Juan R. Diaz, Jose M. Jimenez and Fernando Boronat	- Measurements taken in 2006, show that eDonkey network can have variations of 9 million of users in one hour. - The total amount of data shared inside P2P networks is not dependent with the number of files shared inside the network. - During the experimental period number of files in an eDonkey network has exponentially increased to more than 1billion (the fastest among all P2P networks analyzed). - Also the number of users in eDonkey network increased exponentially to more than 20 million (fastest among the P2P networks analysed). - The average files per user through the years remained almost contant at 100, falling much less in the comparative study with other P2P networks. - All type of files (big files like videos, DVD images) are shared with high correlation between the number of users. More users connected are for 1 - 20 hours. - In the eDonkey is the network with most total amount of shared data inside because many of them are big size files like videos, DVD images, and so on. The total number of users connecting to the P2P filesharing networks is growing. Therefore, the number of users increasing Internet traffic, due to the use of these networks, is growing. - eDonkey in March 2006 had no less than 52 files per user.	Europe	To measure the number of users and, the number of files inside the eDonkey network.	1. The eMule client is modified to analyze the eDonkey network. It is chosen such that it provides the most information on the architecture and the highest update frequency to measure the parameters. 2. Once a desktop file-sharing application joins its network, it receives a message containing the number of users in the network, the number of files shared and, in some cases the total amount of data shared inside the network. 3. This information is periodically refreshed. Using the protocol signature,messages containing these parameters were captured every hour of the day.

Continued on next page

Table 4.4 – continued from previous page

By Whom	Results	Where	What	How
Exploiting semantic clustering in the eDonkey P2P network, S. Handurukande, A.-M. Kermarrec, F. Le Fessant and L. Massoulie.	- For semantically related neighbours, the hit rate is very encouraging (with just 5 peers detected hit rate is close to 30%). - The hit rate when neighbours are selected randomly is very low and remains low even with a significant number of peers. - The hit ratio is influenced by the presence of very generous uploaders who contribute large amount of files, and strongly indicates that the semantic clustering truly exists. - Transitivity of the semantic relationship: Results do not exhibit better results for 2nd level semantic search in particular, if the important factor is the number of peers contacted. However, the fact that 2nd-level semantic search provides similar results as when the same number of semantic neighbours peers are contacted during the first hop, shows that semantic links tend to automatically cluster semantically-related peers. - The hit ratio is negatively impacted when maintaining a peer profile (audio, video etc) per type of file.	Europe	Evaluated several strategies to exploit the semantic proximity between peers against a real trace collected in November 2003 in the eDonkey 2000 peer-to-peer network, to confirm the presence of clustering in such networks and the interest to exploit it.	1. They collected and analyzed peer-to-peer file sharing application trace, focusing on the clustering properties of the peers. 2. They actively probed a community of the eDonkey 2000 clients and obtained a trace of 12,000 clients, sharing 923,000 documents, distributed worldwide with a majority in Europe. The trace contained for each client its list of cache contents. 3. A simple discrete-event simulator composed of n nodes was used to simulate the file exchange between peers and search only using semantic neighbours in a p2p file sharing environment. 4. Nodes are randomly to assigned to the eDonkey clients of the trace. The simulator maintains the global list of the files shared in the system. Each client is associated with a list of files according to the real trace. The simulation consists in, for each client, requesting sequentially each file of its list. 5. They used two different policies (LRU and History based) of maintaining semantic neighbours according to implicit scheme and present a simple explicit scheme; and results were generated.

4.2 Where are P2P Users Located in the Internet?

4.2.1 Introduction

P2P networks have emerged as one of the most prevalent entities on the Internet. These networks allow for large groups of users, employing small, easily available and royalty free, clients to share a vast plethora of resources. Such resources can range from legal content such as Linux distributions to exchange of copyrighted material in the form of songs, movies and software. Such file sharing networks generate a significant amount of traffic when users attempt to share resources among themselves [71]. This is a source for concern to ISPs, since P2P algorithms have been shown to be ISP unfriendly [71] generating large amounts of traffic crossing over inter-AS boundaries, increasing AS-AS traffic and hence resulting in higher operational costs for the service providers.

P2P networks such as Gnutella, Fastrack, Bittorrent (BT), eDonkey, [65, 67, 69, 72] are rampant throughout the Internet today. They are accessed using their vanilla mainline clients and also with a humongous list of their variants. Resources shared among users of such networks are not trivial, either in content, the veracity of which can be gauged by significant legal action against a subset of users of some P2P networks [74]; or in the amount of data that is being transferred to and from clients [68, 70, 73] quietly chugging away. The primary motivation for these networks being: to allow users to share resources effectively and possibly fairly. Naturally, they do not have any consideration for utilizing resources, owned by the ISPs benevolently. It is thereby of utmost importance for ISPs to try and understand the extent of such P2P networks throughout their domains, mainly which ISPs harbor large clusters of users and what methods may be employed to detect such traffic flowing under the hood.

Our research asks the following questions:

1. What kind of network-wide spatial behavior do P2P users display, and which ISPs host large numbers of P2P peers?

2. Which ISPs allow most P2P traffic to pass through their domains?

3. Is the spatial behavior for P2P traffic different from other kinds of traffic, such as http, Internet radio?

We present our research based on profiling P2P flows weaving their way through the ASs in the Internet to understand which ISPs shelter large numbers of P2P users within their domain. This is imperative to understand which ISPs should possibly implement anti-P2P policies more vehemently than others. Additionally, with P2P based content distribution networks becoming a reality [71], this study is even more pertinent to understand which ISPs could cache content for swift delivery to P2P users through these overlay networks. Furthermore, we compare P2P traffic flows with more traditional traffic such as http and Internet radio, based on profiling results, to see if different applications display different network-wide spatial behavior. We slice up the AS structure according to a simple degree based classification, pivoting on CAIDAs AS-degree ranking [79], wherein we label the top 8 ISPs as tier 1, the next 24 as tier 2, the following 48 as tier 3, and the rest as tier 4 since most

ISPs at this level have very few number of connections in comparison to the other ISPs in higher tiers. Each separation point in this simple classification represents a relatively sharp change in AS-degree in the CAIDA dataset, and intuitively differentiates the ISPs among each other. Our contribution can be summed up as follows:

1. We profile over half a million P2P flows, spread over a 30 day period, employing Yahoo DSL and Charter Communications as our primary ISPs for trace collection.

2. We quantify the network-wide spatial behavior of P2P users located in various ISPs, to find that tier 1 [80, 81, 82] and tier 4 ISPs host about 92-98% of all P2P IPs identified from our traces while tier 2 and 3 ISPs seem to hardly host any peers.

3. We identify which ISPs allow large numbers of P2P flows to traverse through their domains, to find again tier 1 and tier 4 ISPs contributing 92-95% of the number of hops on most P2P flows.

4. We profile P2P flows and compare it with other prevalent Internet traffic as http and Internet radio, to ascertain if different applications display different spatial characteristics. We succeed in mining such metrics, such as the IR metric, which may be employed as a first step in conjunction with standard flow identification techniques to home in on suspected P2P traffic.

4.2.2 Where Are My Peers?

P2P peers are distributed throughout the AS hierarchy. We concentrate on ascertaining which ASs host the most end points for P2P flows. For our experiments we chose two popular ISPs, Charter Communications Inc. and Yahoo DSL, from which to initiate connections to various P2P networks. Both these tier 4 providers were chosen for the simple reason that, if we were to choose a tier 1 ISP from which to collect traces we would probably miss out on the spatial behavior displayed by P2P flows as they rise up from lower tiers to tier 1. We would only be able to observe end point distribution but not P2P flow behavior exhibited as the connections traverse towards tier 1 through tiers 2 and 3. We employed a number of clients feeding off Gnutella, FastTrack, and Edonkey networks such as Bearshare, eMule, Limewire, Phex, Gnucleus, Xolox, Kazaa lite, iMesh, and mlDonkey. Traces were collected on 3Mbps links for a period of 30 days and more than half a million P2P flows were analyzed in the process. For trace collection we employed Ethereal as our primary data logging tool, feeding off traces from 22 clients . Custom scripts were used to filter and mine logged data to extract relevant statistics. Lists of popular music files, and videos compiled from well known listings on the Internet [83, 84], were used to inject queries into the P2P network.

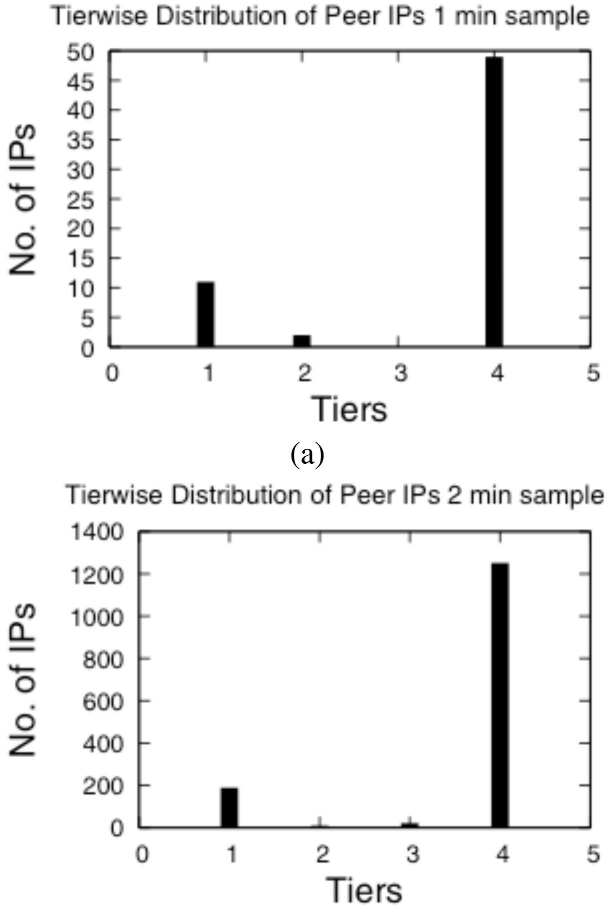
Traces were logged for observation intervals (OIs) of 1, 2, 3, 5, 10, 15 and 30 minutes each. No two OIs for the same or different duration overlap. This was done primarily to determine the temporal robustness of any metrics we develop for comparing P2P versus non-P2P traffic, e.g. to observe if the statistical behavior displayed during a 1 minute OI is the same as displayed within a 5 minute OI. This is critical for developing a robust metric which can be employed for successful identification of P2P flows over a range of observation periods.

We use the latest AS rank data from CAIDA [15], to obtain a complete map of ISPs in the various tiers and employ BGP dumps from [75, 76, 77, 78] for IP to AS lookup. Here we define the end point of a flow to be the final destination IP for that flow and the sink to be the AS at which the flow terminates. We observe a significant percentage of P2P flow end points concentrated in tier 1, and tier 4 ISPs as illustrated in Fig. 4.2 (a) and (b). Fig. 4.3 lists out the end point distribution in the various tiers. We infer, for an observation period ranging from 1 minute to a 5 minute OI, the percentage of tier 1 end points varies from 6.1% to 17.7% of the total number of end points logged for that duration, for P2P flows. Tier 4 ASs consistently contribute a majority of end points, ranging from 79.03% to 87.39%, over the complete range of measurements. The fluctuations in values observed can be related to the fact that with each incrementally increasing OI more P2P peers are contacted in comparison to smaller OI durations, this leads to differences in how many P2P flows end in the various tiers. Surprisingly, ASs in tiers 2 and 3 contribute end points meagerly. For other OIs with durations larger than 5 minutes we observe a similar trend. This skewed behavior is intriguing and poses the following question. Since a large chunk of customers for tier 1 ISPs are large commercial organizations, do these results suggest that large corporate entities may unknowingly be harboring P2P clients on their machines?

We believe that the reason for such skewed statistics are as follows: Most P2P users obtain Internet connectivity via smaller tier 4 ISPs, and it is natural to observe a large concentration of end points in tier 4 ISPs. Some tier 1 ISPs host large numbers of modem based dial-up customers, and sell bandwidth to corporate entities at the same time. We believe that a large part of the contribution from tier 1 ISPs is due to residential customers hooking on via their dial-up connections and joining up with P2P communities. Additionally, we organize end points in bins based on an intuitive sliding scale detailed in Fig. 4.4 and observe the same skewed behavior as displayed by the tier-wise classification. Again, the largest and the smallest of ISPs seem to contribute most significantly to the number of P2P flow end points, displayed in Fig. 4.5. We however do not have a good explanation for why tier 2 and 3 ISPs do not contribute a larger share of P2P end points unlike tier 1 and tier 4 ISPs.

As will be discussed in section IV, this metric, for P2P flows is quite different from non-P2P flows such as Internet radio and http, and may be employed as a low-computation first line of inspection for identifying P2P flows from the huge amount of network traffic generated by a node.

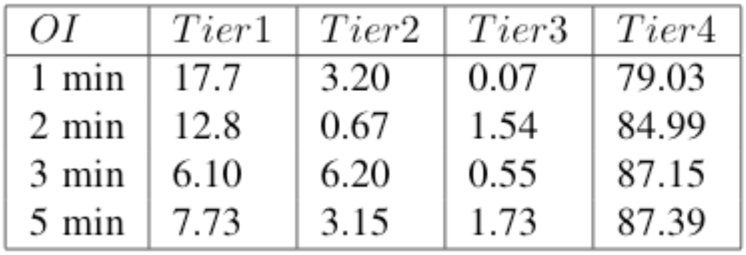
At this juncture we ask, given these statistics would it be prudent to assume that ISPs in tiers 1 and 4 should be the ones to implement anti-P2P policies more vehemently than tier 2 and 3 ISPs?To answer this question, it is imperative to examine which ISPs allow a large number of P2P flows to pass through their domains. This affords us a more informed view regarding which ISPs should perhaps implement anti-P2P policies more industriously than others. Fig. 4.6 and Fig. 4.8 provide an idea of how much transit is provided by ISPs in the various tiers to P2P flows. We say that an ISP provides transit to P2P flows if it allows such flows to pass through its domain. The average number of router hops, distributed tier-wise,



(a)

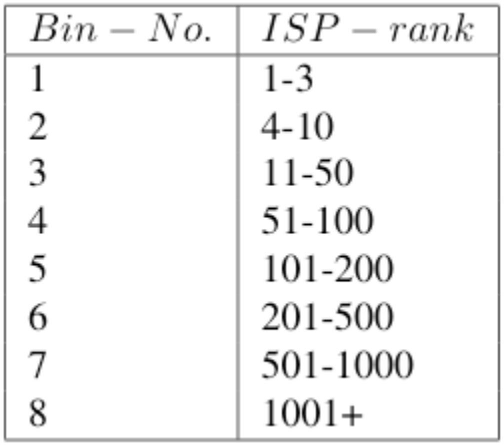
(b)

Figure 4.2: (a) P2P end-point tier-wise distribution for a 1 min trace. (b) P2P end-point tier-wise distribution for a 2 min trace.



OI	$Tier1$	$Tier2$	$Tier3$	$Tier4$
1 min	17.7	3.20	0.07	79.03
2 min	12.8	0.67	1.54	84.99
3 min	6.10	6.20	0.55	87.15
5 min	7.73	3.15	1.73	87.39

Figure 4.3: Tier-wise (percentage) distribution of P2P end-points.



$Bin - No.$	$ISP - rank$
1	1-3
2	4-10
3	11-50
4	51-100
5	101-200
6	201-500
7	501-1000
8	1001+

Figure 4.4: Bin-wise distribution of ISPs according to No. of connections. Data sourced from CAIDA [79].

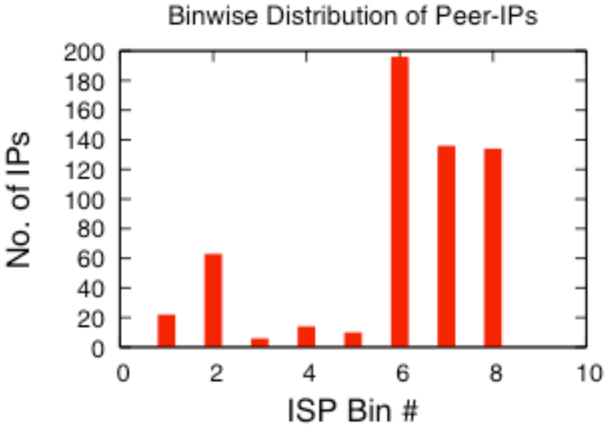


Figure 4.5: Binwise P2P end-point distribution for a 1 min trace.

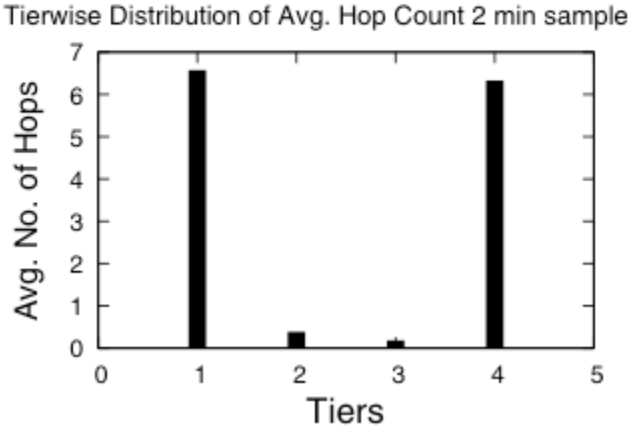


Figure 4.6: Tier-wise distribution of average number of hops of P2P flows, 2 min duration.

OI	$Tier1$	$Tier2$	$Tier3$	$Tier4$
1 min	48.12	5.0	2.5	44.38
2 min	48.70	2.86	1.24	47.2
3 min	45.83	3.6	1.92	48.65
5 min	45.7	5.0	1.91	47.39

Figure 4.7: Tier-wise (percentage) distribution of average number of hops for P2P flows.

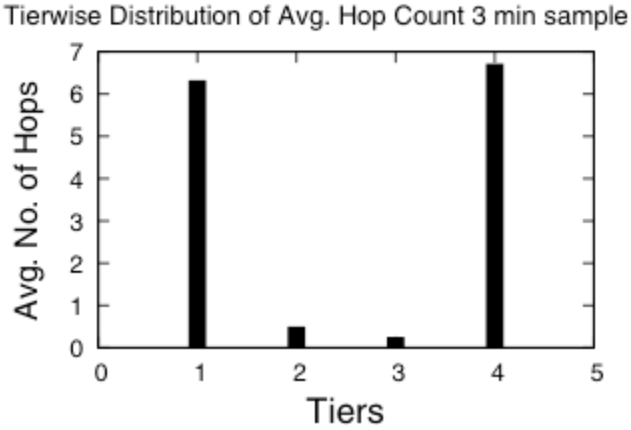


Figure 4.8: Tier-wise distribution of average number of hops of P2P flows, 3 min duration.

for all P2P flows captured during the various time durations provides an insight into which tiers provide more transit than others. Again we observe a skewed distribution, tiers 1 and 4 contain most of the hops in the P2P flows. Apparently, P2P flows seem to traverse through tiers 2 and 3 rapidly while seemingly staying for a longer number of hops in tiers 1 and 4.

This possibly implies that those ISPs which act as large sinks for P2P flows also provide maximum transit for P2P connections. Fig. 4.7 depicts in detail the contribution of each tier in providing transit to P2P flows.

One interesting statistic we observe is that, approximately 98% of all P2P flows traverse tier 1 ISPs and only a very small number of flows do not pass at all through tier 1 ISPs. This alludes towards the hypothesis that tier 1 and tier 4 ISPs not only act as sinks for P2P traffic but also carry most of these flows. This observation suggests that ISPs in tier 1 should implement P2P detection policies hand in hand with tier 4 ISPs. In the following section we compare P2P flows with other kinds of common Internet traffic.

4.2.3 P2P Traffic: A Comparison

In order to further develop an insight into how P2P traffic weaves its way through the AS structure of the Internet we compare it with other forms of prevalent Internet traffic such as http and Internet radio. In this section we present our findings which conclusively prove that that P2P traffic displays a different spatial behavior from these other forms of traffic in the Internet and quantify the characteristics which enable us to differentiate P2P flows from the rest.

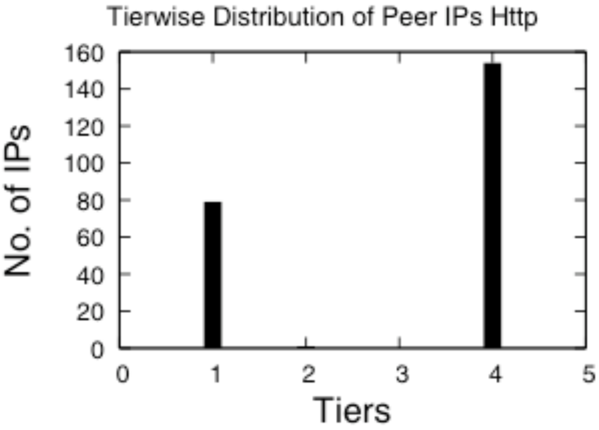


Figure 4.9: Tier-wise distribution of end points of http flows.

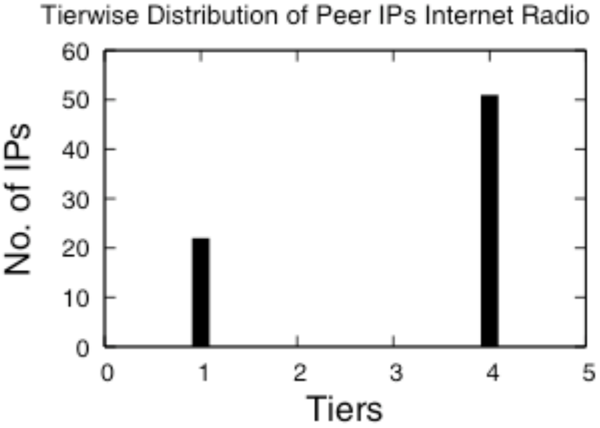


Figure 4.10: Tier-wise distribution of end points of internet radio flows.

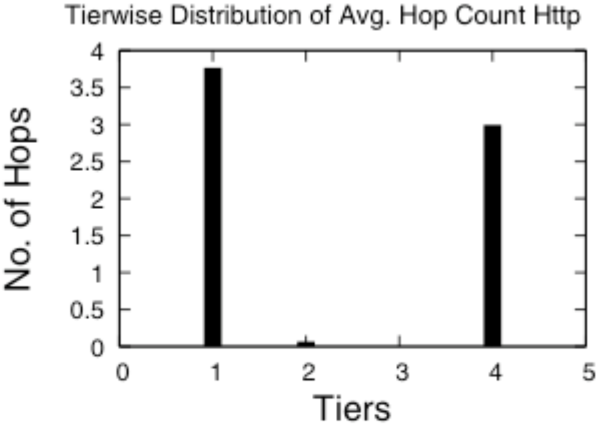


Figure 4.11: Tier-wise distribution of avg. hop count of http flows.

Http and Internet radio traces were captured using Ethereal, running on the same machines with connections through the same ISPs which were used to gather traces for P2P flows. The top 500 websites, compiled from resources on the web, were accessed using automated scripts. Winamp Shoutcast, Yahoo Radio and Real Radio were primary resources for compiling Internet Radio traces. We present Fig. 4.9 and 4.10, detailing out the tier-wise distribution of flow end points of http and Internet radio flows. We observe that this statistic for P2P flows is different from http and Internet radio flows. We define the End-Point-Ratio (EPR) as being the ratio of end points in two tiers, e.g. EPR$_{t1,t4}$ represents the ratio of end points in tier 1 Vs those in tier 4. This provides us with a simple metric with which to compare these traffic flows. EPR$_{t1,t4}$ for Http flows was found to be approximately 0.533, while for Internet Radio applications it was about 0.466. For P2P flows EPR$_{t1,t4}$ varies from approximately 0.0699 to 0.223, significantly different from other kinds of traffic. We also compare how much transit is provided to http and Internet radio flows by ISPs in various tiers of the Internet and compare with statistics obtained for P2P flows. We present Fig. 4.11 and 4.12 which depict the tier-wise average hop count at each tier for http and Internet radio flows. We observe that for P2P flows tiers 2 and 3 provide transit, ranging from 1.2 to 5% of the total number of hops per flow. While for http and Internet radio tiers 2 and 3 contribute a miniscule 0.3 to 1.1%. For http and Internet radio connections only tiers 1 and 4 provide significant transit contributing about 98.9% of the total number of hops for each flow, and for Internet radio about 99.7%. While, for P2P flows, tiers 1 and 4 contribute about 92-95% of all hops per flow. This behavior can be explained by the fact that most popular http sites

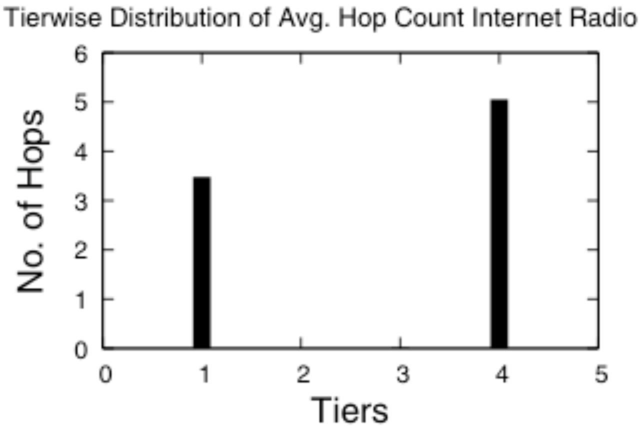


Figure 4.12: Tier-wise distribution of avg. hop count of internet radio flows.

accessed are either cached by local content providers with servers in local tier 4 ISP domains or exist on large high speed content distribution networks as those hosted by the likes of Akamai, a large portion of which possibly resides in tier 1 ISPs. The same could hold true for Internet radio flows.

Additionally, we analyze one more interesting metric, the upslope and downslope of P2P flows versus those of http and Internet radio flows. We define the upslope of a flow as the number of hops needed by a flow to reach the highest tier, from tier 4 to tier 1. Similarly, downslope is simply the number of hops needed by a flow to reach the lowest tier from the highest. This metric, presented in Fig. 4.13, is especially interesting since it suggests that P2P flows traverse a larger number of hops while weaving down the AS hierarchy, from tier 1 to lower tiers as compared to the number of hops needed to reach the topmost tiers, e.g. from tier 4 to tier 1.

Http and Internet radio flows do not display such large imbalance in the number of hops

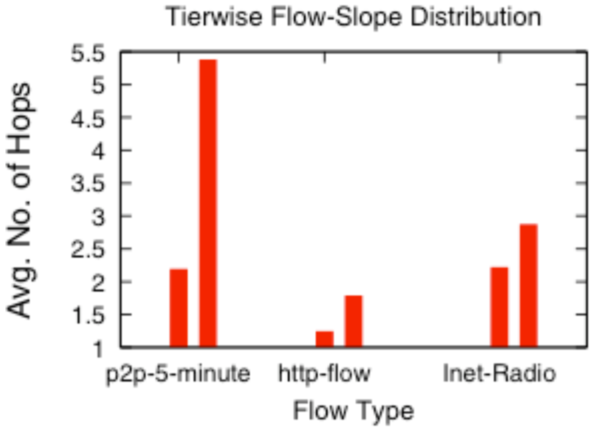


Figure 4.13: Tier-wise distribution of flow slope for P2P Vs http and internet radio flows. Each pair of columns represents up-slope and down-slope for a 5-minute P2P OI, http or Inet Radio. In each pair, the first column represents up-slope and the next one depicts down-slope. Up-slope and down-slope for P2P 2 min and 3 min OIs display similar behavior.

while traversing through the tiers. We define the Imbalance Ratio(IR), as the ratio of number of hops traversed from tier 1 to 4, Vs the number of hops traversed from tier 4 to 1. We observe the IR for P2P flows to range from 1.8 to 2.44, while IR for http flows was observed to be 1.4 and for Internet radio was 1.27. This is a clear differentiation metric between P2P flows and other typesof Internet traffic. Thus adding plausibility to the fact that P2Ptraffic and other prevalent forms of Internet traffic display different network wide spatial behavior. An explanation for such behavior would be, since Internet radio andpopular http sites are hosted on well advertised servers, having high network visibility with entries in most network routers, once a connection reaches a tier 1 ISP it is relatively easy to find a route to the destination server. In case of P2P peers located away from the core of the net, it is but natural to hit larger number of routers in order to find a path to the other peers which definitely have much lesser network visibility than popular servers. In this section we have conclusively proved that P2P traffic displays different spatial behavioral characteristics than other forms of Internet traffic. These metrics can be employed in conjunction with other payload and non-payload based mechanisms to home in on suspect P2P flows for a closer look. In fact since our metrics do not make use of payload sniffing, they are immune to legal ramifications. Further more since we do not link our metrics with specific port based analysis, our mechanism can successfully target P2P clients deliberately using well known ports to mask themselves.

Chapter 5

Podcast based content distribution and P2P

5.1 Introduction

Podcasts are a push-based mechanism for distributing multimedia files such as audio programs or music videos over the Internet. Podcast establishes streams (a.k.a. *feeds*) using either the RSS 2.0 or Atom syndication formats [95] and delivers content for playback on mobile devices and personal computers. The host or author of a podcast is called a podcaster. Podcast enabled web sites may offer direct download or streaming of their content. These content streams are distinguished by their ability to be downloaded automatically using software capable of reading RSS or Atom feeds.

Podcasting is already an important Internet application with roughly 6 million subscribers

[89], and as such, it is an essential component of a complete model of the Interent traffic. Furthermore, podcasting is still growing rapidly towards a projected audience of 56 million by the year 2010 [92], [86], [87]. What started out as a system for distributing homespun radio programming over the Web has now caught on with big media companies. For example, ABC News, NBC News, ESPN, Disney, MTV, FOX, BBC, Apple, CNN and National Public Radio have all introduced podcast programming [92], [93], [94], [95]. Media retail services such as iTunes recently added 3,000 podcast programs to its iTunes online music store. In fact, one of the hubs for subscribing to podcasts, Feedburner.com, manages more podcasts than there are radio stations worldwide [91], and has been recently bought by Google. Further, to provide an idea of how much podcast content traverses the Internet everyday consider the following "conservative" back-of-the-envelope calculation. If 6 million users download an audio file of size 5 MB (a typical size as we see later) per day from only one podcaster, all this content-data amounts to a massive 30 TeraBytes. This number is indicative of the scale of podcast data being transferred and the popularity of this new technology. If we consider the more typical case where podcast listeners subscribe to multiple feeds, the total amount of podcast data can reach hundreds of TeraBytes.

Given its growing trend, we need to model the characteristics of podcasts, especially since podcast distribution differs from other content distribution applications. First, podcasting is a *push*-based distribution [94], [95], and thus it is different from the pull-based approach of web, real-time streaming, youtube-style video. Podcasting pivots on RSS enabled browsers and aggregators [92] which automatically download podcast content [96]. Prefetching of web content has some similarities to push-based approaches, but again it is ultimately user driven, based on popularity and not by *when* content is published by the content provider [112], [111],[113], [114]. Second, high volume websites and streaming video servers are generally hosted by carefully chosen servers, offered by specialized distribution companies like Akamai, with high-bandwith links. In contrast, popular podcast feeds are often home-grown and self-supported endeavors [98], and as such, podcast sources may not be hosted on high-speed servers or in “high-connectivity” network locations, as we discuss later.

In this work, we develop a measurement-based spatial and temporal profile of podcasting, and we use it to develop an open-source modeling tool, SimPod. To the best of our knowledge, this is the first extensive measurement study of podcasting. Our goal is to provide a realistic and comprehensive model that could be used for analytical studies and simulations. For example, our model could be used for network management and provisioning and answer “what-if” scenarios given the growing trend of podcasting. The take away message from our study is that podcast traffic is significantly different from other types of traffic such as web traffic and thus needs to be modeled separately.

Podcasts and P2P : Recently podcasts have been distributed using P2P technology such as BitTorrent [120, 121, 122]. This is a novel way to reduce the bandwidth costs associated with distributing the content. In the absence of P2P technology the publisher of the content often has to reserve a significant amount of bandwidth for distributing the podcast files. With P2P technologies, the bandwidth needed to distribute the podcast files are reduced considerably. Most popular BitTorrent clients now have a podcast feed aggregator built in to them.

Most popular podcasts also offer torrent files for their listeners to download. Consider for example, ”This American Life” a popular show hosted by NPR. It costs $150,000 a year [123], for bandwidth, to distribute this show via podcasts. Using technologies such as BitTorrent could significantly reduce the amount of server-side bandwidth needed by content distributors.

We conduct active measurements, spanning a period of 30 days, from June to July 2006. First, we select 875 podcast streams, from 35 podcasters, based on popularity, according to figures for their subscriber base [90, 91, 96]. We then use PlanetLab to enlist a diverse group of subscribers which subscribe to the selected podcasts and we log their performance. Our main contributions can be summarized in the following points.

a. A detailed profile of podcasting. Based on our measurements, we observe the following interesting characteristics of podcasting.

- **The podcast data profile is significantly different from web/http data:** The average podcast files is approximately 3 orders of magnitude larger than the the average http file. The average and median file sizes are 17 and 22 MB respectively for podcasts files compared to the average http file, which is less than 605KB according to three different studies [99], [100], [101]. In addition, podcast file sizes follow a different distribution, namely a skewed bimodal Gaussian distribution, compared to http files, which follow a heavy-tail Pareto distribution [102].

- **Content is not published uniformly throughout the day:** We observe that US based podcasters sparsely published content during 5AM to 12PM, US-Pacific Time (PST). Popular times for publishing content are 11 PM and 1 AM (PST).

- **Most podcasters publish new content every 5 to 16 hrs:** We observe that the time duration for podcasters publishing new content through respective feeds ranges from 5 to about 220 hrs. Most podcasters display intermission periods of about 5 to 16 hrs in-between publishing new content.

- **The expected content download per podcaster is 2 to 6 MB per day:** We find that a user can expect to download 2 to 6 MB of content per day from a podcaster. The average and median amount of content are 2.5 and 2 MB respectively.

- **Simple caching approaches can provide significant benefits to ASs and users.** Intuitively, this observation can be attributed to the following factors: (a) the spatial distribution of users seems heterogeneous, (b) podcasters are not necessarily located in "central" network places. As a result, we find that caching podcast content in surrounding ASs (one AS away from the podcast source) can reduce the delivery time by nearly 50% for approximately 50% of the users. Moreover, this enables podcast traffic reduction in ASs hosting podcasters as well as load reduction in adjacent ASs.

b. A comprehensive podcast model: SimPod. We synthesize our observations into an easy-to-use podcast model. Our model provides both the qualitative (e.g. distributions of its behavior) and quantitative properties (ranges of values for each parameter), and can be readily used to generate synthetic podcast traffic. SimPod will be made available to the

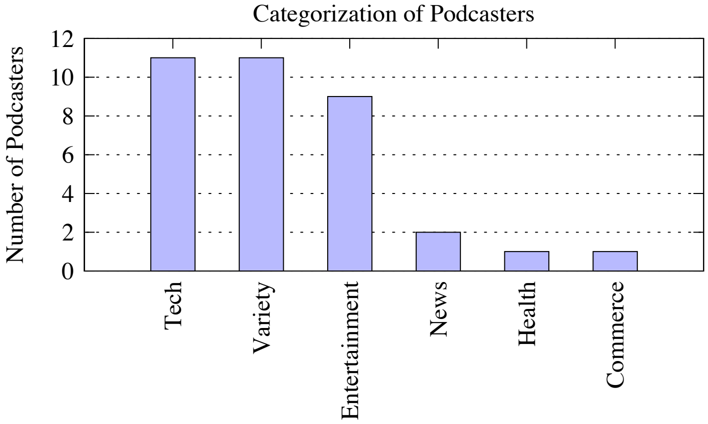


Figure 5.1: Podcaster categories. Tech, variety and entertainment podcasters constitute the majority of podcasts.

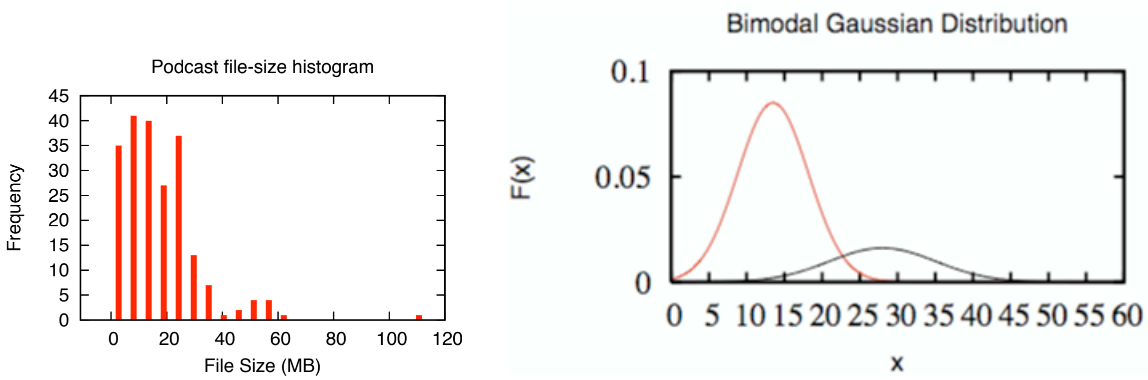
research community as an open-source tool. This traffic generator can generate synthetic podcast traffic which can be embedded into topology-graphs obtained from graph generators as GT-ITM [124] and can generate output which conforms to NS-2's [125] TCL format.

5.2 Data Analysis

We begin with an explanation of the measurement setup.

5.2.1 Experimental Setup

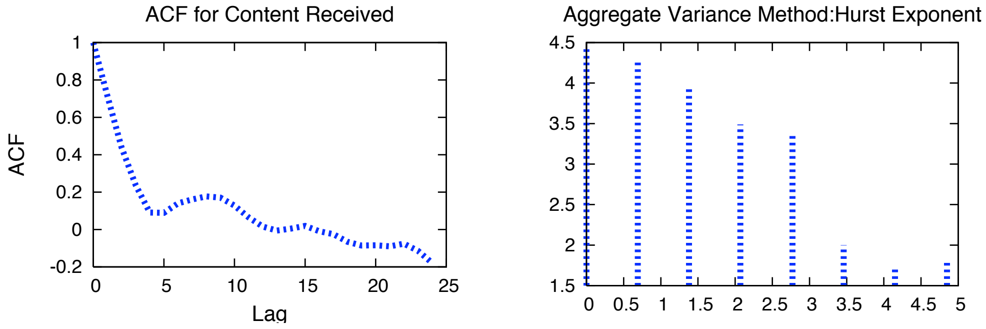
We investigate 35 most popular podcasters (ranked according to number of subscribers) as listed on popular sources on the Internet [90], [92], [93], [94], [95], [96] and initiated connections for 30 days to each of these podcasters from PlanetLab clients. Subsequently, we logged traces of content being streamed from podcasters to the clients. Each podcast client located on PlanetLab nodes queried content servers every 20 minutes (similar to mean polling time



(a) (b)

Figure 5.2: (a) Histogram for podcast file sizes. (b) Bimodal Gaussian distribution, 95% confidence level.

mentioned in [116]) for new content. As soon as new content was detected, log files were updated to reflect temporal statistics. Content was downloaded to measure size and transfer latency. We employed 25 PlanetLab nodes to subscribe to each podcaster. The majority of nodes were spread over the continental US (75%), while others were located in Europe (20%) and Asia (5%). To provide an idea of the kind of content being disseminated by these podcasters, we present Fig.5.1. Podcasters are classified by the various sites [90], [92], [93], [94], [95], [96] into technical, variety, entertainment, news, health and commerce categories. Podcasters in the technical category publish content related to hardware/software news and IT related events. Podcasters in variety category publish content related to current events, family radio shows, lifestyle while those in entertainment category publish music shows and Internet-radio programs. News podcasters publish current events, news reports and sports while those in commerce categories deal with management, investments and shares. Health related podcasters concentrate on general well being.



(a) (b)

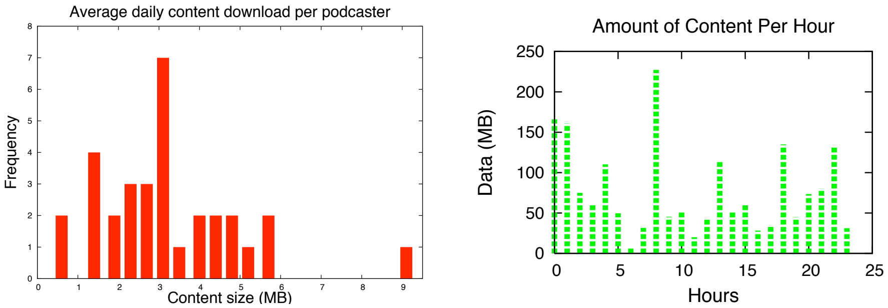
Figure 5.3: (a)Autocorrelation Function for podcast content. ACF values upto lag=10 suggest the presence of memory in the system. Also, negative autocorrelation is observed after this range. (b)Hurst Exponent=0.681, Correlation Coefficient=96.16% (for ordered file sizes).

5.2.2 Data Analysis

Podcast data profile

We first analyze the characteristics of podcast data files. **Observation**: *Podcast content is different from http content*. There are two aspects to this: (a) The type of distribution followed by the file sizes and (b) the average value of the file-sizes.

We present our findings in Fig.5.2.(a) where we plot file size in MB (X axis) versus frequency of files (Y axis) to show the distribution of individual files downloaded from podcasters. We observe that content size downloaded from all podcasters over the complete observation period ranges from 2 to 110 MB. However, 90.6% of files lie within a comparatively smaller range from 2 to 35 MB. *This observation clearly demarcates podcast content from web/http content* since the most probable sizes for podcast data is nearly an order of magnitude larger than average web/http content, about 60 to 605 KB [99], [100], [101], [102]. This is incorporated in SimPod.



(a) (b)

Figure 5.4: (a) Average content download per day per podcaster. (b) Hour-wise Content downloaded from podcasters, over the complete 30 day period.

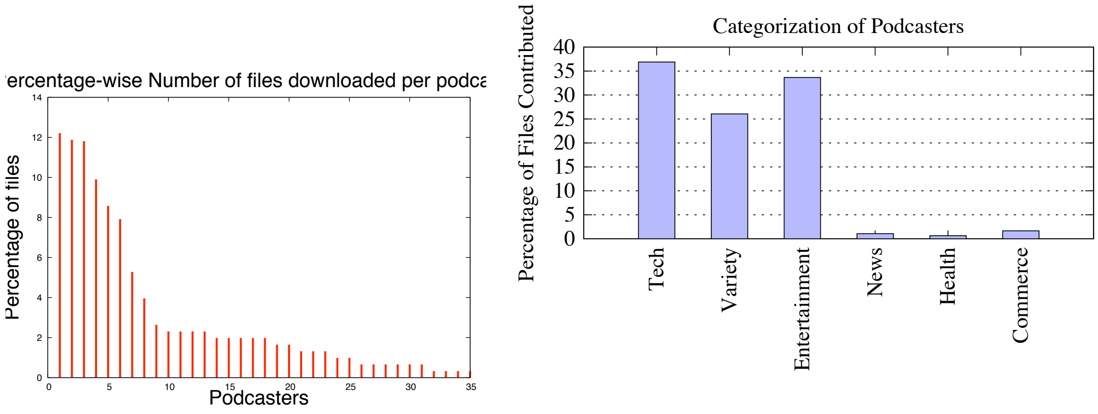
Moreover, we observe that the distribution of file sizes for podcasts conforms to a bimodal Gaussian distribution whose PDF is displayed in Fig. 5.2.(b).

The two Gaussian distributions can be defined by μ=13.5;σ^2=22 and μ=28;σ^2=50. The second distribution contributing the secondary mode observed in the form of a small hump as seen in Fig.5.2.(a). Fig. 5.2.(b) depicts a random sampling of values from a 0 to 1 range from these distributions based on a threshold probability of the bimodal distribution. We observe that for a threshold probability of 0.7, indicating that if a random sample has a lower magnitude, f(x, y)=N(μ=13.5,σ^2=22), else f(x, y)=N(μ=28,σ^2=50), the graph models the decay characteristics of the measured file sizes with less than 5% error. In contrast, file sizes for web/http objects are found to display a heavy tail Pareto distribution [102] which is different from the bimodal Gaussian distribution of the podcast data.

To verify that a unimodal distribution does not effectively model the file-size characteristics we compared the bimodal Gaussian distribution with a pure unimodal γ distribution and find that error rates for unimodal γ are 55.55% worse off than a bimodal Gaussian distribution.

Additionally, we attempt to quantify how much *memory* is present in the file arrival process, i.e., given a particular file size can we predict if the next few files received by the client will be of similar sizes? We present Fig.5.3.(a) which displays the auto-correlation function for the file sizes which are ordered in the manner they were received by clients. Each file is treated as a single sample point. We observe that ACF values upto 10 lags (files) indicate the presence of memory in the file arrival process. Beyond this range we observe negative correlation. We also test for long range dependence in the file arrival process. We present Fig.5.3b, which displays the Hurst parameter (H). It is found to be 0.681, which implies that the file arrival process exhibits long range dependence characteristics. These features are important for modeling purposes and are incorporated in SimPod in later sections.

Observation: *A typical podcaster generates 2 to 6 MB of content per day*. Fig. 5.4.(a), where the X axis depicts content size (MB) versus frequency (Y axis), displays this fact. With podcasting set to garner larger audiences, this metric is significant for ISPs, who want to predict resource demand. Furthermore, end-users can allocate sufficient resources on personal machines to handle daily content downloads. Next, we present Fig.5.4.(b) which shows the total amount of content (over the complete 30 day period) downloaded by a client on an hourly basis. The spikes in the figure point to a large amount of content received during that hour. This data was found to conform to a β distribution with parameters, 2.0 and 24.2. These features are also incorporated in SimPod.



(a) (b)

Figure 5.5: (a) Percentage of number of files contributed by each podcaster with respect to all files downloaded over a 30 day period. 14% of podcasters contribute nearly 54% of files. (b) The percentage of files as contributed by each category of podcasters.

Heterogeneity in podcaster activity

We study the variance in the level of publication activity among different podcasters. We present Fig. 5.5.(a), depicting the percentage of total number of files (Y axis) each podcaster generates during our experiments. The X axis depicts the number of podcasters ranked by the number of files they generate. We observe a skewed distribution: 14% of podcasters contribute over 54% of files, which translates to about 30% of the byte-content. This indicates that a fraction of podcasters are responsible for the majority of content being disseminated. This is expected since certain podcasters host content which is published every few hours while others may not host content or shows which are disseminated as frequently. We find that a bimodal γ distribution, with γ(k=1, θ=2) and γ(k=3, θ=2), models the activity of podcasters, as displayed in Fig.5.6. This is again important for simulating podcast traffic and is incorporated in SimPod. To observe this phenomenon from a coarser granularity, we present Fig.5.5.(b). Clearly tech, variety and entertainment content providers supply the bulk of podcast data received.

5.2.3 Analyzing temporal characteristics

Observation: *Podcast content is published sparsely between 5AM to 12 PM (US-PST).* By performing a temporal analysis of podcast data generation, we ascertain when podcast content is published by podcasters. In Fig. 5.7.(a), X axis depicting the time of day (based on US-PST) and Y axis the frequency of publication of content. We see a timeline for podcast content publication. We observe relatively sedate activity between 5 AM to 12 PM for US based podcasters. This period crudely corresponds to office-hour time on the US east coast. Content is published during other periods of the day although not uniformly. Two clear peaks of publication activity are observed around 11 PM and 1 AM. Also, 3 AM, 2 to 3 PM and 6 PM, seem to be popular times for publication of content. Recall that these observations are averaged over a 30 day observation period. This possibly implies that podcast data is usually published during night hours for dissemination to audiences during the subsequent hours in the morning.

Furthermore, we quantify the delay for publication of new content by a podcaster, which we will refer to as **inter-file** delay, in Fig. 5.7.(b). Again the X axis depicts time in hours, and Y axis the frequency. Clearly, a 5 to 16 hour inter-file delay period seems to be most prevalent. Also, approximately 58 and 112 hour inter-file delay also seem to be common. 112 and 58 hour inter-file delays could possibly correspond to shows that are broadcast once

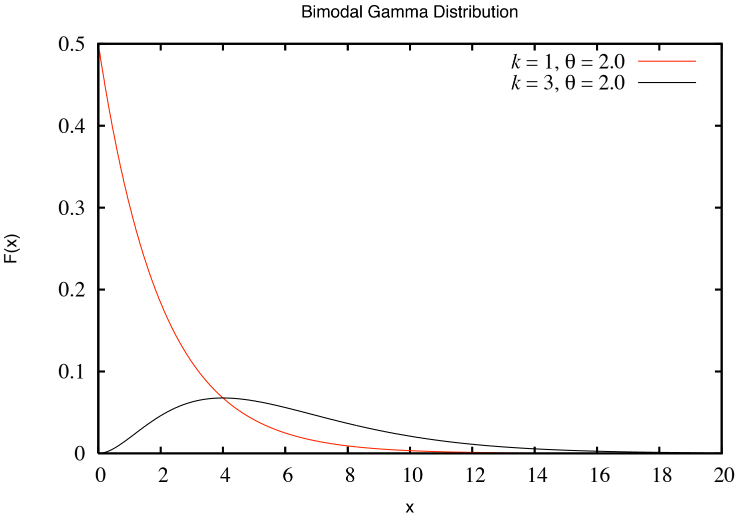
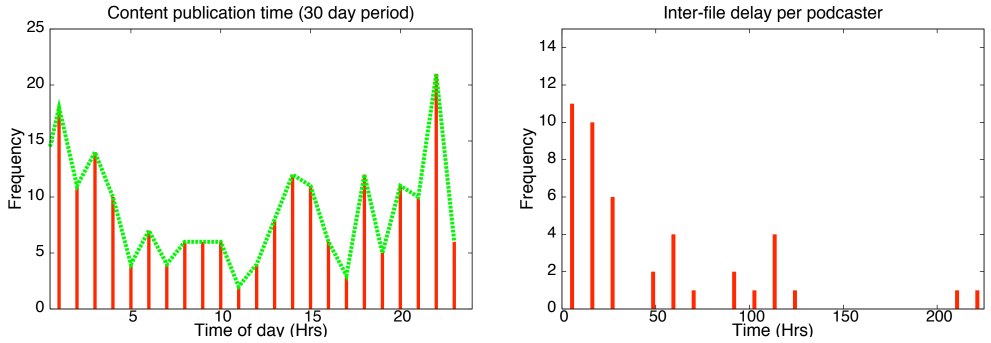


Figure 5.6: Bimodal γ distribution which successfully reproduces the characteristics heterogeneity amongst podcasters for percentage of files contributed. The combined distribution can described by γ(k=1, θ=2) and γ(k=3, θ=2). 95% confidence level.

or thrice a week respectively. This metric is important to understand the nature of podcast flows and is incorporated in SimPod. Information such as a 5 hour inter-file delay can help ISPs understand the impact this kind of traffic as it passes through their networks.

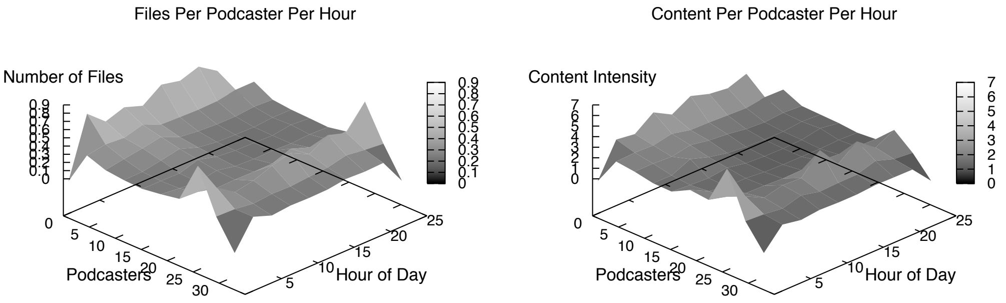
We present a different view of the temporal analysis of podcasters in Fig.5.8. In Fig.5.8.(a) we observe files received from each podcaster in every one hour slot. In Fig.5.8.(b) we observe content received from each podcaster in the same one hour slot. From Fig.5.8.(a) we observe two consistent peaks running through the 24 hour spectrum. This provides insight regarding heterogeneity of podcasters. Further, in Fig.5.8.(b) we observe similar peaks running through the 24 hour spectrum again.

In the subsequent section we focus on specific characteristics of podcast flows as they



(a) (b)

Figure 5.7: (a) Timeline for podcast content publication by podcasters, on a 24 hour scale. (b) Inter-file delay per podcaster, in hours. The most common inter-file delay ranges from about 2 to 16 hours.



(a) (b)

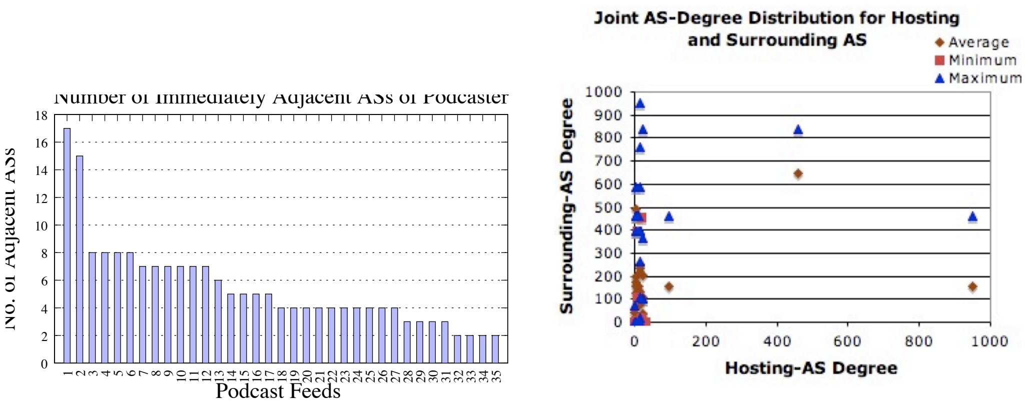
Figure 5.8: (a) Number of files sent per-podcaster, per-hour. X axis depicts the 35 podcasters, the Y axis depicts 0-24 hour timescale, while the Z axis depicts the normalized number of files sent by each podcaster during that time slot(b) Amount of content (Bytes) sent per-podcaster (normalized), per hour. Similar definitions hold true for axes.

move through different ISP domains.

5.3 Flow Analysis

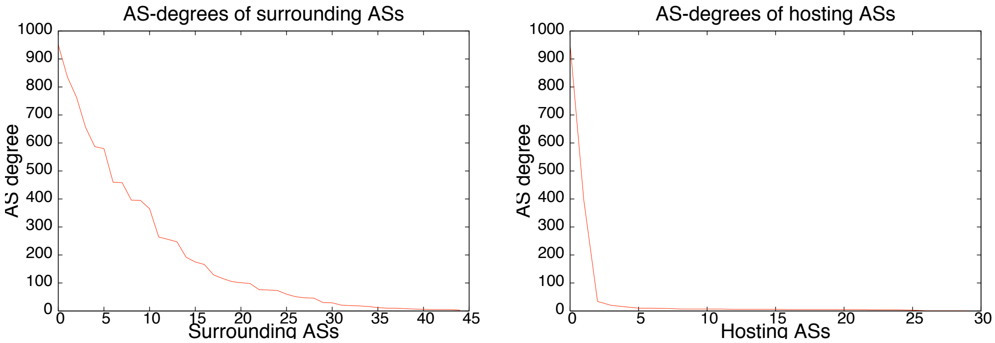
In this section we investigate the **spatial characteristics** of podcasters. We aim to quantify:

1. Location of the podcasters in the Internet.



(a) (b)

Figure 5.9: (a) Number of surrounding ASs of each podcaster. Nearly 70% of podcasters have less than 7 or more surrounding ASs. (b) Joint AS-degree distribution for hosting and surrounding ASs



(a) (b)

Figure 5.10: (a)AS degree for surrounding ASs.(b)AS degree for hosting ASs. Clearly the majority of hosting ASs are situated in smaller ISPs in comparison to surrounding ASs.

2. Profiling network-paths from podcasters to clients.

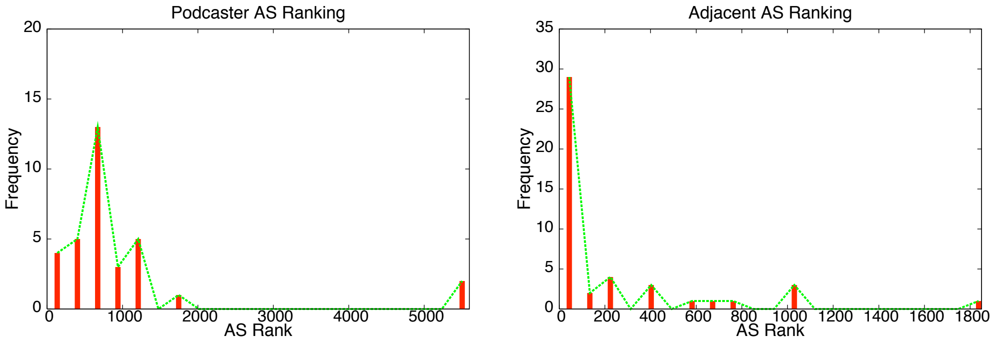
3. Benefits for ASs and end-users if content-caching were employed.

5.3.1 Where are the podcasters located

We refer to ASs in which podcasters are physically located as **hosting ASs** and those ASs which exchange traffic directly with hosting ASs as **surrounding ASs**.

Observation: *A large portion of hosting ASs have less than 7 surrounding ASs.* We observe from Fig. 5.9.(a) that about 70% of podcasters have less than 7 distinct surrounding ASs. Only 17% of hosting ISPs display connections to 8 or more surrounding ASs. These statistics are important for selecting which nodes in a synthetic topology should be designated as podcasters and they are incorporated into SimPod. Further, in Fig. 5.9.(b) we represent the joint AS-degree distribution for hosting and surrounding ASs. Clearly hosting ASs which have low AS-degrees are connected to surrounding ASs with much higher AS-degrees. This is depicted by the cluster observed near the Y-axis.

We refer to the **rank of an AS**, based on the information made available by [108], which assigns a rank to an AS according its degree. However, note that the degree of an AS correlates with the ”importance” and role in the hierarchy of the ASs: i.e. the top 10 highest degree ASs are the top 10 tier-I providers. We observe that nearly 40% of hosting ASs are ranked near 650 to 1000, according to the latest CAIDA [108] dataset. This is displayed in Fig.5.11.(a). Other popular AS ranks for hosting ASs seem to range around 134 to 402, and 900 to 1254, contributing nearly 27 and 24% respectively. Clearly these figures imply that a majority of hosting ASs do not have high AS-degrees. Also, we find that 84.4% of all surrounding ASs were found to have degree based AS-ranks [108], based on the number of connections, higher than 500, as displayed in Fig.5.11.(b). In Fig. 5.10.(a) and (b), we show the AS degrees of the various surrounding ASs and hosting ASs. This is significantly different from metrics obtained for hosting ASs. This implies that **surrounding ASs are usually of higher degree compared to the hosting ASs**. In fact this leads us to believe that

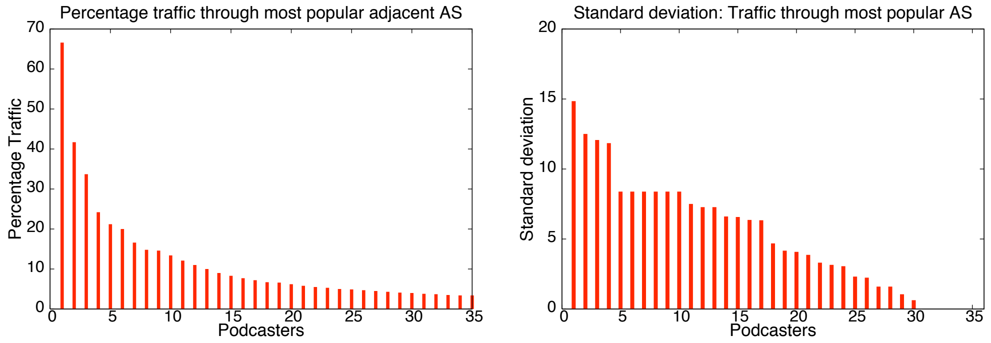


(a) (b)

Figure 5.11: (a)AS-ranking of hosting ASs which house podcasters. (b) AS-ranking of surrounding ASs which allow podcast subscription requests and possibly content to pass through them.

hosting ASs are possibly customers of most surrounding ASs. This observation gains special significance when we discuss whether it makes sense for surrounding ASs to employ caching mechanisms for podcast content.

In our attempt to verify if podcasters are different from web/http content providers, we analyze top 35 websites [119], to find that 67.8% of them have AS-ranks less than 500 and only 10.7% lie between AS-ranks 900-1290. This leads us to believe that **web/http content providers are located in ASs with higher degrees than podcasters**. Another important observation is that most incoming requests for podcast data do not pass uniformly through all surrounding ASs. In fact we observe a significantly skewed distribution for the same, depicted in Fig.5.12. These characteristics are captured while simulating podcasts using SimPod.



(a)

(b)

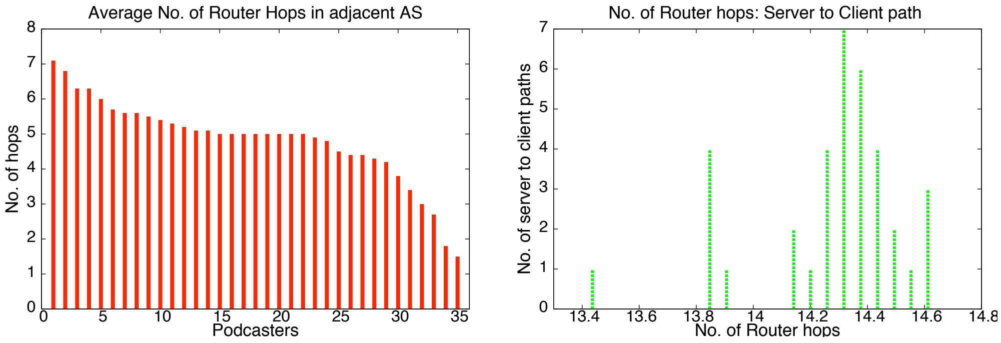
Figure 5.12: (a) The percentage of podcast traffic which passes through surrounding ASs, for each podcaster. (b) The standard deviation of the percentage of traffic passing through surrounding ASs. A comparatively high standard deviation for nearly 28.5% of podcasters suggests that popular surrounding ASs receive a large share of traffic, and would possibly benefit from caching podcast content.

5.3.2 Profiling network-paths from podcasters to clients

As mentioned previously, **Observation**: *Podcast content does not pass uniformly through all surrounding ASs*. In Fig. 5.12.(a) we observe that 17% of podcasters see 20 to 66.6% of their traffic passing through one (the most popular) surrounding AS. The most popular AS being one of the surrounding ASs which allows most of the connections initiated from our clients to pass through itself onwards to the hosting AS. Another important observation we highlight here is that standard deviation for amount of traffic passing through the most popular AS in comparison to amount traffic passing through other surrounding ASs, per podcaster, is comparitively high for nearly 28.5% of podcasters. This is depicted in Fig.5.12.(b). This fact will prove useful when we discuss where to cache podcast content.

Here we refer to router-hops whenever we mention *hops* in the remainder of this work. We present Fig.5.13, which depicts the average number of router hops a podcast flow would incur in a surrounding AS. We observe that 85% of podcast flows incur 4 to 7 router hops in surrounding ASs. This is significant, considering that most content flows from podcast servers to clients incur about 14 router hops. This is depicted in Fig. 5.13.(b). Cutting down podcast flows from meandering through routers located within surrounding ASs would definitely be of interest to ISPs since it would obviously reduce podcast traffic load within their domains. Fig.5.14.(b) gives an idea of the worst case, in terms of number of router hops experienced by podcast flows moving through surrounding ASs. In fact the worst case number of hops for most podcasts is found to be worse off by nearly 70 to 100%. This clearly implies that the majority of podcast content flows incur large number of router hops in surrounding AS's in comparison to the other distinct AS's traversed while passing from server to client. This is a good motivation for surrounding AS's to consider caching schemes inside their domains to reduce this skewed router-hop statistic. Shortening the number of hops would be beneficial not only for surrounding AS's but also for the end-users who would observe a reduction in the overall content-path length.

Comparison with HTTP sites: The number of router hops it took for traceroutes to reach the 35 web/http websites [119] from our clients was approximately 10. This suggests that popular web/http content providers are located in top-tier networks so that clients can reach them without incurring large delays. To reach podcasters our clients incur a larger number of hops simply because podcast data is not hosted on extremely well-connected edge servers. This could occur primarily due to the socio-economic reasons driving publication of podcast content. Podcasters do not always publish content for financial benefit, quite a few podcasters wish to share their daily experiences, views on current events and entertainment with listeners



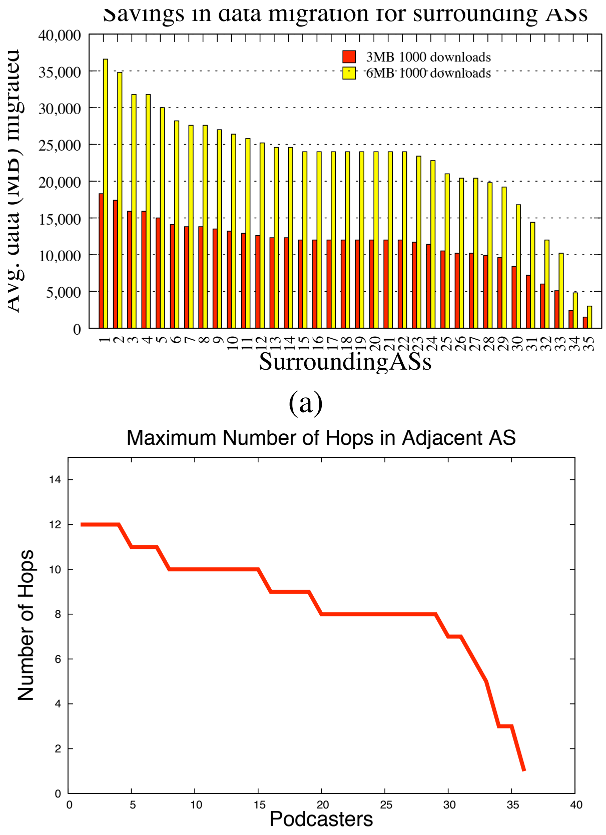
(a) (b)

Figure 5.13: (a)Average number of router hops in surrounding ASs of each podcaster incurred by podcast flows. (b) Distribution of average number of router hops as seen by paths from podcaster to client.

for free. This is hardly ever the case with popular websites. Successful web/http content providers usually subscribe to a profitable business model and thus can afford the services of content distribution networks.

5.3.3 Quantifying benefits of caching

Here we use the term *caching* to depict content-caching from the servers point of view. The question that we wish to raise are: *how much benefit do ASs and users stand to gain if podcast content is cached*?. To answer this question we need to analyze spatial characteristics of of those ASs through which podcast feeds pass through. Hence, we initiate traceroutes to find paths between podcasters and end-users. We know that Internet routing is asymmetric and we discuss this limitation in detail later in section VI. We use PlanetLab nodes to contact podcasters since we do not have a complete database of podcast subscribers. Our attempts to contact podcasters in the hope of gaining access to audience data were unsuccessful. Hence, the only option was to use clients under our control and then observe the spatial characteris-



(a)

(b)

Figure 5.14: (a) Savings in the amount of content(MB) which needs to be migrated through a surrounding AS. The red bars represent the amount of data surrounding ASs would have to transfer within their domains, from one router to another, if 3 MB of daily content is downloaded by 1000 users per day. The yellow bars represent the same case for 6 MB daily content downloaded by 1000 users. (b) Worst case: Maximum number of router hops in surrounding ASs of each podcaster incurred by podcast flows.

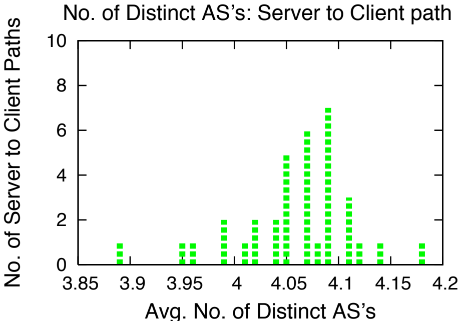


Figure 5.15: Distribution of average number of distinct AS's as seen by paths from podcasters to clients.

tics of podcast flows.

In this section our aim is to highlight the inherent incentives for ISPs to cache podcast content and not to focus on which CDNs should podcasters use. To implement content caching, a plethora of mechanisms are available. The two main goals of all these mechanisms is (i) reduce traffic load on server and (ii) bring content closer to end-users for faster access. To meet these objectives content can be cached either near or at the server to satisfy the former goal, or near the clients, to satisfy the latter one. As we will see not only are hosting ASs and end-users the ones to gain from caching podcast content but surrounding ASs too benefit from reduction in podcast traffic load through their domains, assuming that caching policies are implemented at the borders. This is a definite motivator for ISPs who allow podcast feeds to pass through their domains to implement podcast content caching.

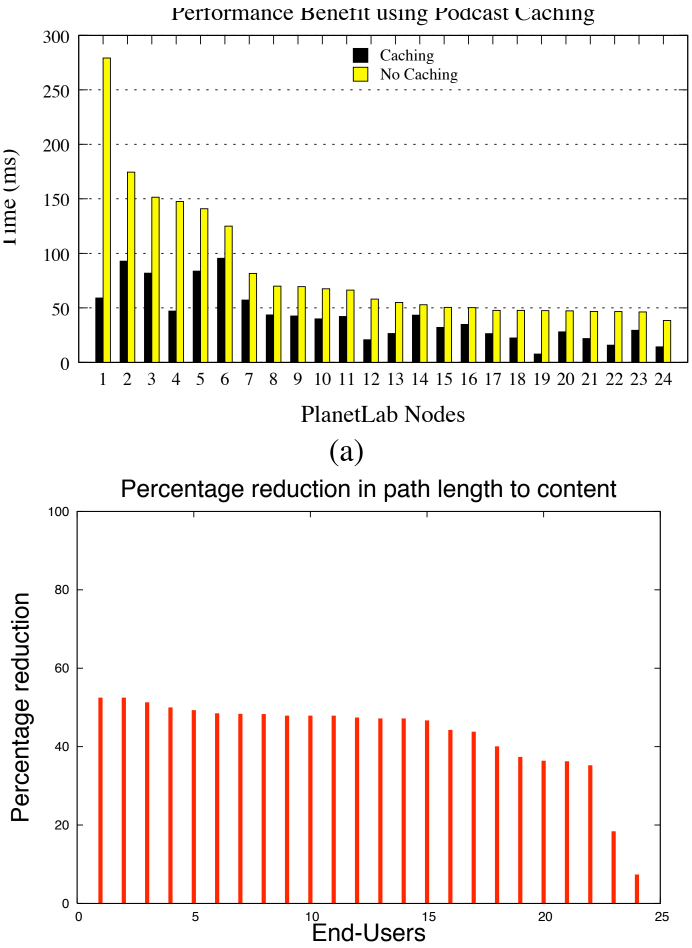
We observe in Fig.5.14.(a), the amount of data that surrounding ASs could avoid transferring from one router to another as users make requests for podcast content is significant. We consider 1000 users requesting content per day,which is a very conservative estimate given that some popular feeds such as WNYC and New York Public Radio have audience numbers bordering 16,000 listeners. Savings on traffic for 1000 subscribers a day ranges from 2.5 to 67 GB per podcaster. Reducing this kind of traffic load would be beneficial for any ISP.

Observation: *at least 50% percent of clients see a reduction in content access time by nearly 50%.* We present Fig.5.16.(a), highlighting benefits seen by end-users if content is cached by surrounding ASs. This is a significant improvement in speed of content delivery, and is one of the strongest motivators for caching podcast content in surrounding ASs.

Furthermore, we observe from Fig.5.16.(b) that a significant reduction in path length is experienced by end users while attempting to access content cached in surrounding ASs. In fact the path length reduction for nearly 88% of end-users ranges from 35 to 52% of the original path length. This not only reduces the content access time as seen previously but also reduces total network load for podcast content access. Our additional efforts to characterize the complete path from podcast server to clients is represented in Fig. 5.15 and 5.13.(b). Clearly, the most prevalent number of distinct AS's seen by paths from hosting AS's to clients is 4. This gives an idea of the number of entities involved in the transmission of these content flows. Caching policies implemented to efficiently deliver content to end-users benefits all these enities.

5.4 SimPod: Simulating Podcasts

In previous sections, we have highlighted characteristics of podcast data and podcaster location. Primarily how podcasters differ from more traditional http content providers and this gives rise to the need for modeling this mode of content distribution. Podcasts are different not only in terms of content size from web data, but also with respect to temporal and spatial respects. We use our analysis in previous sections to develop SimPod. We begin by categorizing important features of podcasts into two base classes: **location** and **data**. Within each class we describe how to develop a simulation model, which displays behavior similar to our data traces. In Fig. **??**, we provide an overview of our model.



(a)

(b)

Figure 5.16: (a) Performance benefit employing caching of podcast content in surrounding ASs. End-users experience a significant reduction in content delivery latency. (b) Percentage reduction in path length for end-users when podcast content is cached in surrounding ASs.

5.4.1 Data

This class defines the workload characteristics of a podcast simulation. File sizes follow a bimodal Gaussian distribution with parameters (13.5, 22) and (28, 50). This information encompasses deviation between file sizes for realism. To simulate unequal behavior of podcasters, the podcaster relative-activity metric, which is easily implemented as a single bit allowing a particular podcaster to publish a file in an even driven simulation can be drawn from a bimodal γ mixture with shapes 1,2 and 3,2 respectively. Now, we address the time of publication and find that it should be drawn from a Gaussian distribution with parameters 8.7 and 4.6. Further, we can simulate the inter-file intervals between publishing content according to a Gaussian distribution with parameters 61.14 and 52.5.

5.4.2 Location

In this class we provide information regarding *where should podcasters be placed* given a BGP-level topology. Podcaster AS ranks are displayed in Fig. 15. These AS ranks should be generated from a β distribution with shape parameters 0.66 and 5.15 to satisfy the finer properties listed in previous sections. Also, ASs surrounding these hosting ASs should have ranks drawn from a γ distribution with shape 0.35. Similarly surrounding AS degree and hosting AS degree should follow a β and a Gaussian-kernel based distribution as listed in Fig. **??**. Further, the number of surrounding ASs should conform to a Poisson distribution with parameter 5.48.

SimPod allows users to select clients according to their topological specifications. We do not restrict the user and provide only general guidelines for client selection from a synthetic toplogy based on metrics we have seen in previous sections. SimPod can use features such as number of router-hops in surrounding ASs, number of distinct ASs on network-path and number of router hops from podcaster to client to select clients. Users can specify their own criteria to select subscribers.

5.4.3 File generation process

Here we present the analysis of the podcast file generation process. We test the measured file arrival process for stationarity in order to estimate parameters for synthetic modeling. This is imperative for successful Auto Regressive Moving Average [115] modeling. Applying the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [115] we can uncover if the incoming file process is level stationary or not. We find p value for the test to be 0.1 with KPSS level at 0.273, implying that the stationarity hypothesis is true. To further substantiate this observation we apply the augmented Dickey-Fuller test [115] on the incoming file process to confirm stationarity. We find the p value to be 0.01 with the Dickey-Fuller level at -4.357. This bolsters the claim that the file arrival process as seen by end-users is stationary. Now, we provide details of the ARMA (1,1) model to describe the file arrival process in Table. 5.1. An ARMA (1,1) model is formally described in the following manner: $y_t = a_0 + a_1 y_{t-1} + b_1 e_{t-1}$. Where y_t represents the numeric vector or time series to be fit into the ARMA model and a_0 represents the intercept, while a_1 and b_1 are the estimated coefficients. The error variable is represented by the e_{t-x} series. The first two rows of Table. 5.1 define the

Table 5.1: Coefficients and residuals for the ARMA modeling of the file generation process: Podcaster to client.

Min	$1Q$	$Median$	$3Q$	Max
-22.4528	-1.6312	-0.7010	0.7498	26.9933
Coefficients	Estimate	Std. Error	t value	Pr($>\|t\|$)
ar1	0.60996	0.06603	9.238	<2e-16
ma1	0.22086	0.08168	2.704	0.00685
intercept	2.62598	0.64550	4.068	4.74e05

statistics of the file size data received by a client. The following four rows depict estimates for the coefficients. The last column displays the ”significance level” of each coefficient, all being below 0.05 which proves the efficacy of the model. Other similar models such as ARMA (1,2), ARMA(2,1) and ARMA(2,2) were not found to produce statistically significant estimates of coefficients.

SimPod will be made available as an open source simulation framework for the reseach community. SimPod can place podcasters and clients on topologies generated by GT-ITM and can generate traffic patterns which are similar to NS2-TCL format, this provides a painless interface with popular network simulation tools.

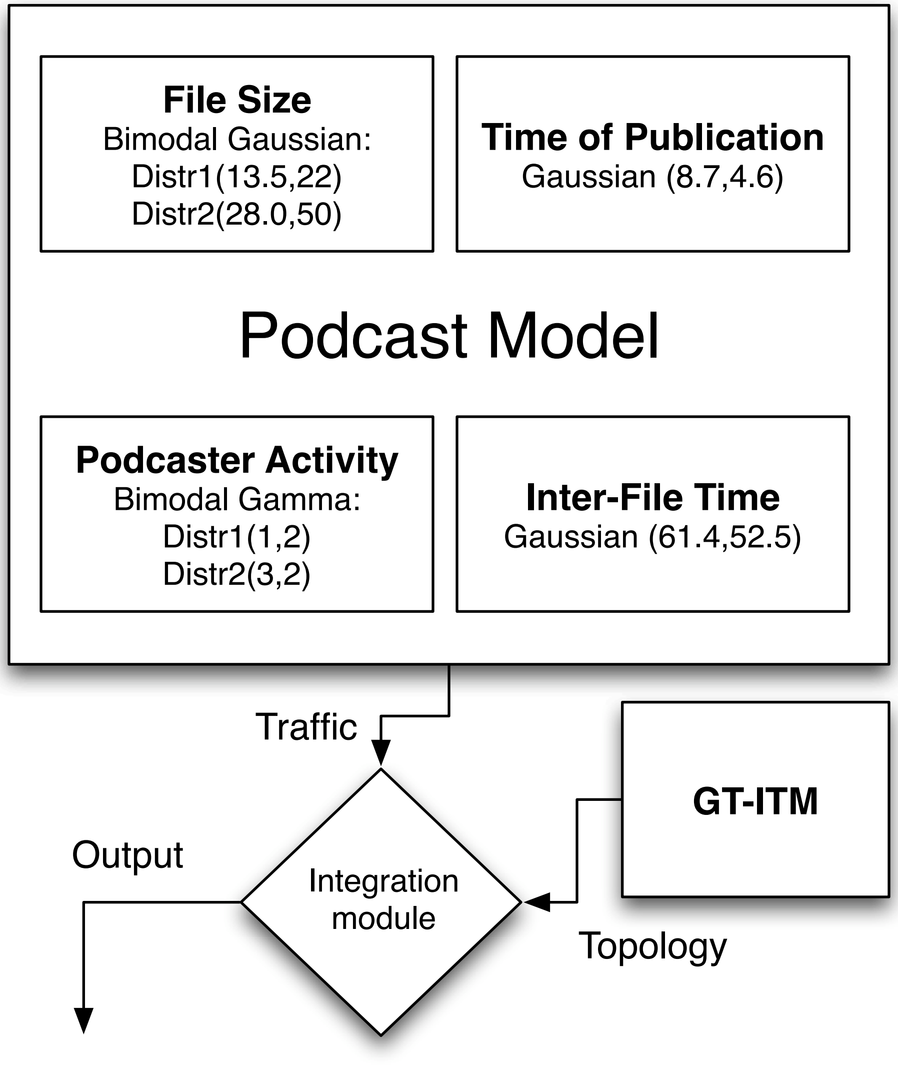


Figure 5.17: SimPod : Internal modules and integration with GT-ITM topology generator.

Chapter 6

Conclusion and Future Work

In this chapter we summarize the various contributions we have made via the research presented in this thesis. We mention the main take-away points which highlight the unique contribution of these research efforts.

6.1 Privacy Concerns in Peer-to-Peer (P2P) networks

To the best of our knowledge, this section contains work which is the first to quantify the probability that a user will be tracked by blocklisted IPs, and thus, potentially run the risk of a lawsuit. Using Planetlab, we conduct large-scale active measurements, spanning a period of 90 days, from January to March 2006, spread over 3 continents, yielding over a 100 GB of TCP packet header data. We find that **a naive user is practically guaranteed to be contact blocklisted IPs:** we observe that 100% of our peers run into blocklisted users. In fact, 12% to 17% of all distinct IPs contacted by a peer are blocklisted IPs. Interestingly, a little caution can have significant effect: the top five most prevalent blocklisted IPs contribute to nearly 94% of all blocklisted IPs we ran into. Using this information users can reduce their chances of being tracked to just about 1%. At the same time, we examine various different dimensions of the users such as the geographical location and the role of the node in the network. We find that the geographical location, unlike the role, seems to affect the probability of encountering blocklisted users. Finally we examine, who are the blocklisted IP addresses. Interestingly, we find that 0.5% of all distinct IPs belong explicitly to media companies. The major of the blocklisted users seem to belong to commercial and government organizations and a sizeable portion of the most popular belong to anonymous BOGON ranges.

Our work is the first step in monitoring the new phase of "wars" between the content providers and the P2P community. It will be very interesting to continue to monitor the evolution of this conflict. For example, one could analyze the accuracy and completeness of the blocklists, and the speed with which a new blocklisted entity is flagged.

6.2 Measuring Spatial Properties of P2P content streams

6.2.1 Analyzing P2P-network measurement approaches

In this section we have compiled extensive information about the the eDonkey network and its characteristics. This report delves into compiling what the research community knows about the eDonkey network and its interesting features. Clearly, the number of users of this P2P network seems to be on the increase. The number of distinct files being shared on this network can be as high as 11 million. We have found that reports about the size of the network seem to be disparate, ranging from a 230,000 users to 9 million users. Further, nearly 25-36% of users in the eDonkey network operate from behind a NAT. Most eDonkey users share less than 10 files each and the number of free riders can be as high as 84% of the total population. Additionally, we also discuss the methodologies followed by the various research efforts and highlight their potential and limitations.

6.2.2 Where are P2P Users Located in the Internet?

Our research clearly highlights the skewed distribution wherein a majority of P2P flows end at tier 1 and tier 4 ISPs to the tune of 92 to 98%of all P2P flows analyzed. Also, 92 to 95%of P2P flows traverse through tier 1 and tier 4 ISPs, incurring a larger number of hops in these tiers than in tiers 2 and 3. Furthermore, tier 2 and tier 3 ISPs do not seem to participate significantly in providing transit to P2P traffic neither do they act significantly as sinks for the same. Interestingly, a considerable percentage of P2P flows, nearly 98% of the complete observation set, managed to reach tier 1 ISPs and weave through their domains. These facts may encourage tier 1 and 4 ISPs to implement anti-P2P policies more vehemently than others. Moreover, we observe that P2P flows traverse a larger number of hops while weaving down the AS hierarchy, e.g. from tier 1 to tier 4 as compared to the number of hops needed to reach the top most tiers from the lower ones. The imbalance metric referring to this observation, in conjunction with others developed throughout this section conclusively prove that network-wide spatial behavior displayed by P2P flows is very different from other forms of prevalent

Internet traffic.

6.3 Podcast based content distribution and P2P

Through our research we have shown that file-sizes for podcast content follow a bimodal Gaussian distribution. This is different from http content which follows heavy tail Pareto distributions. We show that podcasters are located in smaller ASs with lower degrees than ASs which host popular web/http content providers. Most podcast files range from 2 to 35 MB in size. Further analysis reveals results which hold significance for ISPs, such as users subscribing to popular podcasts could expect to download 2 to 6 MB of content per day per feed from a podcaster.We observe that all podcasters are not equally active, about 14% of podcasters contribute over 54% of files and this trend can be mimicked by a bimodal γ function. Also, via a time based analysis, we find that podcasts are sparsely published sparsely during 5AM to 12PM. This information is important for ISPs who wish to predict traffic loads on their infrastructure. Additionally, we find that the most common inter-file delay for a podcast feed is about 5 to 16 hours. By analyzing AS-rank information of hosting and surrounding ASs, we are able to deduce provide-customer relationships between hosting and surrounding ASs. Based on this observation, implementation of simple caching mechanisms in surrounding ASs, which have peering or customer-provider relationships, with hosting ASs can help cut expected latency by about 50%. Additionally, this leads to at least 17% of ISPs hosting podcasters, reducing podcast traffic load by 20 to 66.6%. We also provide a detailed simulation model for synthetic podcast traffic generation. Given the rising status of podcasting, it is essential to monitor and model these content flows since they are bound to play an important part in the future Internet.

Bibliography

[1] http://narus.com

[2] http://www.sourceforge.net

[3] http://www.mediadefender.com

[4] http://peerguardian.sourceforge.net

[5] http://www.sandvine.com/news/pr_detail.asp?ID=203

[6] Electronic Frontier Foundation, Packet Forgery By ISPs: A Report on the Comcast Affair., Nov. 2007.

[7] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim *A Survey and Comparison of Peer-to-Peer Overlay Network Schemes*, IEEE Comm. Survey and Tutorial, March 2004.

[8] http://news.dmusic.com/article/7509

[9] http://www.betanews.com/article/MPAASuesUsenetTorrentSearchSites

[10] http://importance.corante.com/archives/005003.html

[11] http://www.mp3newswire.net/stories/napster.html

[12] http://news.com.com/2100-1027-995429.html

[13] http://sourceforge.net/projects/peerprotect

[14] http://bluetack.co.uk/blc.php

[15] http://www.boycott-riaa.com/article/9316

[16] http://slashdot.org/articles/02/05/25/0324248.shtml

[17] http://www.planet-lab.org

[18] T. Karagiannis, A.Broido, M. Faloutsos, and kc claffy, *Transport layer identification of P2P traffic*, In ACM Sigcomm IMC'04, 2004.

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

[20] S. Sen and J. Wang, *Analyzing Peer-to-Peer Traffic Across Large Networks*, In ACM SIGCOMM IMW, 2002.

[21] Thomas Karagiannis, Pablo Rodriguez and Dina Papagiannaki, *Should Internet Service Providers Fear Peer-Assisted Content Distribution?*, In IMC'05, Berkeley, 2005.

[22] Kurt Tutschku, *A measurement-based traffic profile of the edonkey filesharing service*, In PAM'04, Antibes Juan-les-Pins, France, 2004.

[23] http://www.techspot.com/news/16394-record-labels-launch-legal-action-against-kazaa.html

[24] http://www.mpaa.org/CurrentReleases/2004_12_14_WwdeP2PActions.pdf

[25] Valerie Alter, Building Rome in a Day: What Should We Expect from the RIAA?,56 HASTINGS COMM. & ENT. L.J. 155.

[26] Jane Black, The Keys to Ending Music Piracy, BUS. WK., Jan. 27, 2003, http://www.businessweek.com/bwdaily/dnflash/jan2003/

[27] RIAA Gives Advance Warning to Song-Swappers Before Lawsuits are Filed, http://www.antimusic.com/news/03/oct/item77.shtml, 2003.

[28] Thomas Karagiannis, Andre Broido, Nevil Brownlee, KC Claffy, Michalis Faloutsos, *Is P2P dying or just hiding*, IEEE Globecom 2004.

[29] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan, *Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload*, Proc. of SOSP-19, October 2003.

[30] Chu, J., Labonte, K., and Levine, B. N., *Availability and locality measurements of peer-topeer file systems*. In Proc. of ITCom '02.

[31] F. Clvenot-Perronnin and P. Nain, *Stochastic Fluid Model for P2P Caching Evaluation*, In Proc. of IEEE WCW 2005.

[32] http://azureus.sourceforge.net/plugin_details.php_plugin_safepeer

[33] http://mutella.sourceforge.net/

[34] http://www.billboard.com/bbcom/charts/chart_display.jsp?fT̄he_Billboard_Hot_100

[35] http://www.mp3hits.com/charts/euro

[36] http://www.trustyfiles.com

[37] http://isc.sans.org/diary.php?date=2005-04-11

[38] http://www.winmxworld.com/tutorials/block_the_RIAA.html

[39] http://xeex.com

[40] http://www.completewhois.com/bogons/index.htm

[41] http://phoenixlabs.org

[42] http://www.mtvasia/Onair

[43] http://ed2k.2x4u.de

[44] en.wikipedia.org/wiki/EDonkey

[45] G.Siganos, S.L. Tauro, M.Faloutsos, Jellyfish: A Conceptual Model for the AS Internet Topology, Journal of Communications and Networks 2006.

[46] Marios Iliofotou , P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, G. Varghese, Network Monitoring Using Traffic Dispersion Graphs (TDGs), In ACM SIGCOMM Internet Measurement Conference (IMC), San Diego, CA, 2007.

[47] Kurt Tutschku, A Measurement-Based Traffic Profile of the eDonkey Filesharing Service, Passive and Active Network Measurement, 2004, Vol. 3015, Pages 12-21.

[48] Wenjie Wang and Cheng Jin and Sugih Jamin, Network Overlay Construction under Limited End-to-End Addressability, CSE-TR-489-04, EECS Department, University of Michigan, 2004.

[49] F. Fessant and S. Handurukande and A. Kermarrec and L. Massoulie, Clustering in peer-to-peer file sharing workloads, In Proceedings of the 3rd International Workshop on Peer-to-Peer Systems (IPTPS) (2004).

[50] S. B. Handurukande and A.-M. Kermarrec and F. Le Fessant and L. Massoulié and S. Patarin, Peer sharing behaviour in the eDonkey network, and implications for the design of server-less file sharing systems, ACM SIGOPS Oper. Syst. Rev., Vol. 40, No. 4, 2006, Pages 359-371.

[51] Wenjie Wang and Hyunseok Chang and Amgad Zeitoun and Sugih Jamin, Characterizing Guarded Hosts in Peer-to-Peer File Sharing Systems, In Proceedings of IEEE Global Communications Conference, 2004, Vol. 3, Pages 1539-1543.

[52] Nicolas Christin and Andreas S. Weigend and John Chuang, Content availability, pollution and poisoning in file sharing peer-to-peer networks, EC '05: Proceedings of the 6th ACM conference on Electronic commerce, 2005, Pages 68-77.

[53] Jia Yang and Hao Ma and Weijia Song and Jian Cui and Changling Zhou, Crawling the eDonkey Network, GCCW '06: Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops, 2006, Pages 133-136

[54] Walid Saddi and Fabrice Guillemin, Measurement Based Modeling of eDonkey Peer-to-Peer File Sharing System, Managing Traffic Performance in Converged Networks, 2007, Vol. 4516, Pages 974-985.

[55] J. Lloret and J.R. Diaz and J.M. Jimenez and F. Boronat, Public Domain P2P File-Sharing Networks Measurements and Modeling, International Conference on Internet Surveillance and Protection, 2006

[56] S. B. Handurukande and A.-M. Kermarrec and F. Le Fessant and L. Massoulié, Exploiting semantic clustering in the eDonkey P2P network, EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop, 2004.

[57] CISCO Systems Inc., The Exabyte Era, White Paper, 2008.

[58] Wayne Lai, Managing Peer-To-Peer Applications in Dormitory Networks, White Paper, 2004.

[59] Fivos Constantinou and Panayiotis Mavrommatis, Identifying Known and Unknown Peer-to-Peer Traffic, NCA '06: Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications, 2006, Pages 93-102.

[60] Gerhard Hablinger and T-Systems and TU Darmstadt, Peer-to-Peer Networking and Traffic Management on Internet Platforms, Tutorial - 13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems, 2006.

[61] Mo Zhou and Yafei Dai and Xiaoming Li, A measurement study of the structured overlay network in P2P file-sharing systems, In Journal of Adv. MultiMedia, 2007, Vol. 2007, No. 1, Pages 10-10.

[62] Thomas Karagiannis and Andre Broido and Michalis Faloutsos and Kc claffy, Transport Layer Identification of P2P Traffic, In Proceedings of ACM Sigcomm Internet Measurement Conference, 2004.

[63] Subhabrata Sen and Oliver Spatscheck and Dongmei Wang, Accurate, scalable in-network identification of p2p traffic using application signatures, WWW '04: Proceedings of the 13th international conference on World Wide Web, 2004, Pages 512-521.

[64] A. Kothari and D. Agrawal and A. Gupta and S. Suri, Range Addressable Network: A P2P Cache Architecture for Data Ranges, P2P '03: Proceedings of the 3rd International Conference on Peer-to-Peer Computing, 2003.

[65] B. Cohen, Incentives build robustness in Bittorrent, In Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, 2003.

[66] Ethereal. http://www.ethereal.com/.

[67] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. A. Felber, A. Al Hamra, and L. Garces-Erice, Dissecting BitTorrent: Five Months in a Torrents Lifetime, In PAM04, Antibes Juan-les-Pins, France, 2004.

[68] T. Karagiannis, A. Broido, M. Faloutsos, and kc claffy, Transport layer identification of P2P traffic, In ACM Sigcomm IMC04, Taormina, Italy, 2004.

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

[70] S. Sen and J. Wang, Analyzing Peer-to-Peer Traffic Across Large Networks, In ACM SIGCOMM IMW, 2002.

[71] Thomas Karagiannis, Pablo Rodriguez and Dina Papagiannaki, Should Internet Service Providers Fear Peer-Assisted Content Distribution?, In IMC05, Berkeley, 2005.

[72] Kurt Tutschku, A measurement-based traffic profile of the edonkey file-sharing service, In PAM04, Antibes Juan-les-Pins, France, 2004.

[73] Thomas Karagiannis, Dina Papagiannaki and Michalis Faloutsos, BLINC: Multilevel Traffic Classification in the Dark, ACM SIGCOMM, Philadelphia, PA, USA, August 2005.

[74] http://www.techspot.com/news/16394-record-labels-launch-legal-action-against-kazaa.html

[75] http://www.ripe.net

[76] http://www.lacnic.net

[77] http://www.afrinic.net

[78] http://www.arin.net

[79] http://www.caida.org

[80] Lixin Gao, On Inferring Autonomous System Relationships in the Internet, IEEE/ACM Transactions on Networking (TON), Volume 9, Issue 6 (December 2001). Pages: 733 745.

[81] Ramesh Govindan and Anoop Reddy, An Analysis of Internet Inter-DomainTopology and Route Stability, In INFOCOM 1997.

[82] L. Subramanian, S. Agarwal, J. Rexford and R. H. Katz, Characterizing the Internet Hierarchy from Multiple Vantage Points, In IEEE Infocom 2002.

[83] http://www.billboard.com/bbcom/charts/chart display.jsp?f?TheBillboardHot 100

[84] http://www.mtvasia/Onair

[85] http://news.com.com/

[86] http://www.tdgresearch.com/

[87] http://news.bbc.co.uk/1/hi/technology /4658995.stm

[88] http://www.paulcolligan.com/2006/06/20

[89] http://www.edisonresearch.com/home/archives/LATimes060405.pdf

[90] http://www.podcastalley.com/

[91] http://blogs.feedburner.com/feedburner/archives/00175.html

[92] http://www.podcast.net/

[93] http://www.ipodder.org/

[94] http://www.podcastingnews.com/archives/2005

[95] http://en.wikipedia.org/wiki/Podcast

[96] http://www.podcastdirectory.com/podcasts/

[97] http://images.apple.com/education/solutions /podcasting

[98] http://www.smbtrendwire.com/2007/

[99] http://www2.sims.berkeley.edu/research /projects/how-much-info-2003/internet.htm - wbsamp

[100] http://www.dlib.org/dlib/april03/ lavoie/04lavoie.html

[101] http://www.pantos.org/atw/35654.html

[102] M. Arlitt and C. Williamson, Internet Web Servers: Workload Characterization and Performance Implications, IEEE/ACM Transactions on Networking , Vol. 5, No. 5, Oct. 1997.

[103] Kun-Lung Wu, Philip S. Yu and Joel L. Wolf, Segment-Based Proxy Caching of Multimedia Streams, Procs. WWW10, '01.

[104] Jia Song, Segment-based proxy caching for distributed cooperative media content servers, SIGOPS Op.Sys. Rev, vol.39, 05.

[105] Li Fan, Pei Cao, Jussara Almeida and Andrei Broder, Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol, Procs of ACM SIGCOMM'98, pp. 254-265.

[106] A. Banerjee, A. Mitra and M. Faloutsos, Dude where's my peer, Globecom 2006.

[107] Thomas Erlebach, Alexander Hall, and Thomas Schank: Classifying Customer-Provider Relationships in the Internet. IASTED Intl. Conf.on Comm.'s and Comp. Networks (CCN 02), pages 538-545, Nov'02.

[108] http://www.caida.org

[109] Y. He, M. Faloutsos, S. V. Krishnamurthy, Bradley Huffaker. "On Routing Asymmetry in the Internet", IEEE GLOBECOM 2005 - Autonomic Internet, St Louis, MO, USA, Nov 2005.

[110] Y. He, M. Faloutsos, S. V. Krishnamurthy. "Quantifying Routing Asymmetry in the Internet at the AS Level", IEEE GLOBECOM 2004 - Global Internet and Next Generation Networks, Dallas, Texas, USA, Nov 2004.

[111] V. N. Padmanabhan and J. C.Mogul, Using predictive prefetching to improve World Wide Web latency, Sigcomm 96.

[112] A. Bestavros and C. Cunha, Server-initiated document dissemination for the WWW, IEEE Data Eng. Bulletin, Sep. 1996.

[113] M. Crovella and P. Batford, The network effects of prefetching, Procs. of Infocom 1998.

[114] Jia Wang, A Survey of Web Caching Schemes for the Internet, ACM Computer Communication Review (CCR), Vol. 29, No. 5, October 1999.

[115] A. M. Gun, M. K. Gupta and B. Dasgupta, An outline of statistical theory, The World Press, 2003.

[116] V. Ramasubramanian, R. Peterson and Emin Gun Sirer, Corona: A High Performance Publish-Subscribe System for the World Wide Web, in Procs. of NSDI 2006.

[117] http://www.podtrac.com

[118] K. J. Christensen and N. J. Javagal, Prediction of future world wide web traffic characteristics for capacity planning, Int. J. of Netw. Manag., Vol. 7, No. 5, 1997.

[119] http://www.alexa.com/site/ds/top_500

[120] http://geekswithblogs.net/jemimus/archive/2004/12/11/17304.aspx

[121] http://weblogs.asp.net/cfranklin/archive/2005/01/17/354911.aspx

[122] http://www.bittorrent.com/search?q=podcast

[123] http://surana.wordpress.com/2008/06/17/need-bittorrent-for-itunes/

[124] Ellen W. Zegura, Kenneth Calvert and M. Jeff Donahoo. A Quantitative Comparison of Graph-based Models for Internet Topology. IEEE/ACM Transactions on Networking, December 1997.

[125] http://www.isi.edu/nsnam/ns/