Macro- and Microscopic Analysis of the Internet Economy from Network Measurements

Thesis by:
Jakub Mikians

Advisor: Pere Barlet-Ros
Co-Advisor: Nikolaos Laoutaris
Co-Advisor: Josep Solé-Pareta

PhD Program: Arquitectura i Tecnologia de Computadors
Department of Computer Architecture
Universitat Politècnica de Catalunya - BarcelonaTech

March 2015

For Ania, Maja, my parents and my brother.

Abstract

The growth of the Internet impacts multiple areas of the world economy, and it has become a permanent part of the economic landscape both at the macro- and at microeconomic level. On-line traffic and information are currently assets with large business value. Even though commercial Internet has been a part of our lives for more than two decades, its impact on global, and everyday, economy still holds many unknowns.

In this work we analyse important macro- and microeconomic aspects of the Internet. First we investigate the characteristics of the interdomain traffic, which is an important part of the macroscopic economy of the Internet. Finally, we investigate the microeconomic phenomena of price discrimination in the Internet.

At the macroscopic level, we describe quantitatively the interdomain traffic matrix (ITM), as seen from the perspective of a large research network. The ITM describes the traffic flowing between autonomous systems (AS) in the Internet. It depicts the traffic between the largest Internet business entities, therefore it has an important impact on the Internet economy. In particular, we analyse the sparsity and statistical distribution of the traffic, and observe that the shape of the statistical distribution of the traffic sourced from an AS might be related to congestion within the network. We also investigate the correlations between rows in the ITM. Finally, we propose a novel method to model the interdomain traffic, that stems from first-principles and recognizes the fact that the traffic is a mixture of different Internet applications, and can have regional artifacts. We present and evaluate a tool to generate such matrices from open and available data. Our results show that our first-principles approach is a promising alternative to the existing solutions in this area, which enables the investigation of what-if scenarios and their impact on the Internet economy.

At the microscopic level, we investigate the rising phenomena of price discrimination (PD). We find empirical evidences that Internet users can be subject to price and search discrimination. In particular, we present examples of PD on several e-commerce websites and uncover the information vectors facilitating PD. Later we show that crowd-sourcing is a feasible method to help users to infer if they are subject to PD. We also build and evaluate a system that allows any Internet user to examine if she is subject to PD. The system has been deployed and used by multiple users worldwide, and uncovered more examples of PD.

The methods presented in the following work are backed with thorough data analysis and experiments.

Contents

List of Figures

List of Tables

Acknowledgements

We gratefully thank DANTE, the GÉANT network operator, for kindly providing us with the data used in Part I. We would also like to thank CESCA for allowing to collect the data used in this work. We acknowledge *ipoque* for kindly providing access to their PACE traffic classification engine.

Jakub Mikians was funded by FI Grant 2010FI_B 00512 from Generalitat de Catalunya. The research was supported by the following projects and institutions:

- NOMADS project, Spanish Ministry of Science and Innovation (ref. TEC2011-27474)
- SUNSET-B: Sustainable network infrastructure enabling the future digital society, Spanish Ministry of Economy and Competitiveness (ref. TEC2014-59583-C2-2-R)
- Comissionat per a Universitats i Recerca del DIUE of the Catalan Government (ref. 2009SGR-1140)
- Comissionat per a Universitats i Recerca del DIUE of the Catalan Government (ref. 2014SGR-1427)

All the research and data presented in this work was conducted prior and independently to my employment in Amazon.

List of Acronyms

AS Autonomous System

BGP Border Gateway Protocol

CDN Content Delivery Network

CP Content Provider

DSL Digital Subscriber Line

GDP Gross Domestic Product

ICT Information and Communications Technology

ISP Internet Service Provider

ITM Interdomain Traffic Matrix

IXP Internet Exchange Point

NTP Network Time Protocol

OD Origin-Destination, in a context of pairs of network nodes

PD Price Discrimination

POP Point of Presence

TM Traffic Matrix

TP Transit Provider

UPC Universitat Politècnica de Catalunya

VPN Virtual Private Network

Chapter 1: Introduction

The past decades witnessed the advent of the Internet and its evolution from a research network to a web spanning across the world, interconnecting furthest points of the globe. The Internet initiated or catalysed changes in many areas of human activity and in parallel it heavily affected the global economy. With almost 40% of world population online [43], the economy of the Internet accounts for one fifth of the global GDP growth in recent years [56]. The Internet reshaped the economic landscape in many areas. It also affected the traditional, pre-Internet industry. It is estimated that even 75% of Internet economic impact comes from traditional industries [56]. For those companies the Internet became a new advertising channel, a new sales channel or a new way to manage business, but also increased their exposure to global competition. However the Internet is not merely a tool accelerating traditional economy. It created a new digital economy, with economical phenomenas reflecting its unique nature. Although the size of the "Internet economy" is hard to estimate, its share in global GDP lies between 3.4% and 4.1% [56, 13]. If the Internet was a national economy, it would rank in the global top five [13].

Some of the biggest Internet market players, like AT&T or Comcast, were present on the Information and Communications Technology (ICT) field for years and fit naturally to the new environment. Others companies, like Google or Amazon, are children of the digital economy, and provide services that could not exist without the network. Interactions between those highest level players, their Internet business policies and the impact they have on network traffic, shape the Internet macroeconomic landscape.

On the other side of this economic ecosystem there is the regular user who creates demand for connectivity, content, and for various services. For a large user base, Internet is an important place of work, retail and social interaction [46]. At the same time, the user generates a wide spectrum of personal information. This information, accumulated from the bulk of users is a valuable resource on its own and is desired by network marketing companies and online retailers. Interactions between the individual users, retailers and service providers contribute to the Internet economy at micro-scale. Even though those two viewpoints seem distant, cumulative decisions of the bulk of unpredictable users can instantaneously change the flow of the revenues. For instance, users that switched from phone carrier messaging services to online messaging applications, had taken away $23 billion in revenue from carriers in 2012, and $33 billion in 2013 [16, 18]. This shows how important is to investigate economic phenomenas at both macro- and micro-scale together.

In this thesis we take a look at the Internet economy from two different perspectives. First, from the macro-scale standpoint, we examine the traffic flowing between Autonomous Systems (AS). This is the highest level of communication in the Internet, where Internet Service Providers transfer bulk data through their infrastructure. At this level, a single user is not visible. Instead, large scale patterns and phenomenas are observable, which allows a researcher to ask important questions about evolution of the Internet [29] or network neutrality [84]. In this work we characterize traffic between ASes, and later propose a method to generate synthetic traffic matrices that could be useful in simulations and in evaluating different what-if scenarios. As the revenue of the biggest stakeholders on the Internet is directly related to traffic volumes, understanding characteristics of the traffic and being able to model it brings us closer to understand the macroeconomics of the Internet.

Later, we look at the Internet economy from the micro-scale point of view and we investigate microeconomic phenomenon at the intersection of areas of personal information and retail business. Namely, we look at the Internet economy from a perspective of a regular user and explore the issue of price discrimination. We look for empirical evidence that this well known economic phenomenon [65] exists on the Internet and present a feasible and scalable approach that can help Internet users to determine if they are subject to price discrimination. We show that private information is also part of a wide and growing economic landscape [64].

Conducting data driven experiments requires excellence from the researchers in handling large quantities of data. In addition to the previously described research areas, and as a fruit of our initial exercises with processing bulk backbone data, we present a novel algorithm to analyse backbone traffic on-line that allows to detect malicious portscan activity. The experience gathered during this exercise became the basis of the tools developed and used in the rest of the thesis to process massive amounts of network traffic. This work is presented as appendix of this thesis.

1.1 Motivation and problem statement

Internet economics has many facets and can be analysed at different levels, as discussed in the previous section. In this section we present the motivation and challenges behind our research on interdomain traffic and price discrimination, as two important aspects of the Internet macro- and microeconomics.

1.1.1 Macroscopic view

At its highest level the Internet is organized into Autonomous Systems, where an "AS is a connected group of one or more IP prefixes run by one or more network operators which has a single and clearly defined routing policy." [41]. In most cases it is a network operating on a large area (e.g., metropolitan, countrywide or worldwide) under the government of a single organization. At this level of Internet, the flow of the money is directly related to the flow of the traffic. The better the knowledge about the traffic ASes have, the better peering decisions they can make to be ahead of the competition. Not surprisingly, detailed information on traffic volumes is considered to be a sensitive business information, and ASes do not reveal such data publicly.

Despite of the dependence between Internet macroeconomics and the interdomain traffic, there is little knowledge about the properties of the latter. The main obstacle for the researchers in this area is the scarcity of publicly available data. This creates a demand for insight into properties of the interdomain traffic. Some invaluable works present different aspects of the interdomain traffic [30, 33, 49, 74, 79].

To this end, we aim to infer statistical properties of the Interdomain Traffic Matrix. In our work in Chapter 2 we use passive NetFlow data from the European-wide GÉANT network. We are aware that it depicts a small, and biased, fraction of the interdomain traffic, but still it is one of the most complete data sets currently available to the researchers.

Besides the knowledge of properties of the interdomain traffic, researchers and network operators are interested in modelling the traffic. For instance, they would like to know how the traffic will change when the user base changes, an application starts being popular in a particular geographical region or when a new popular application emerges.

A natural next research step would be to create a model that could produce synthetic interdomain traffic matrix with specific properties. Given the scarcity of data, such a model would be useful in Internet macroeconomics research, as it would allow to generate a synthetic, but representative traffic matrix of an arbitrary size, and would allow to evaluate various what-if scenarios.

There exist several methods to infer some information about interdomain traffic. For instance, [34] presents a methodology to infer traffic from CDN logs and [23] proposes a method to infer invisible elements of the traffic matrix. According to our best knowledge, the trailblazing work of Chang *et al.* [26] is the only work presenting a full approach to generate a synthetic traffic matrix. The authors of [26] use a mixture of "utilities" and attribute the traffic to the considered AS types. In contrast, we would like the model to reflect application-level characteristics of the traffic. As discussed in the Introduction, popular applications used in large scale (e.g. messaging applications, peer-to-peer file sharing, file hosting services, video streaming) impact interdomain traffic, and thus Internet macroeconomics, directly. Also, a macroscopic view on the Internet economy would require from a model to recognize that different applications can generate different traffic patterns in different geographical regions.

To this end, in Chapter 3 we strive to create a model that allows to generate a synthetic traffic matrix. The model is based on first-principles and allows to include different applications (*e.g.* web, P2P) by recognizing that those applications have different forward and reverse traffic ratios. It also models differences in regional popularity of the applications. Eventually we use this model to discuss a what-if scenario, where we include "cloud storage" application in the model. We believe that the above-mentioned traits of our model will make it applicable in the area of the Internet economy.

1.1.2 Microscopic view

On the other end of the Internet economic pyramid there are regular users who request content, use online services and interact with the other users. All those activities generate network traffic which is handled by the service providers. At the same time, each user's move in the network leaves a small chunk of information giving a hint about himself. Services and applications that collect those bits are often accessible for "free", with a stipulation that the user, in exchange to the possibility of using a service, will share his personal information. This became the prevalent business model in the Internet and for the first time the vast amount of data about users behaviour is accessible so easy and on such large scale.

This automatically raises privacy concerns. The Internet allows profiling of the end users on a scale without any precedence in history. The profiling information on the users activity is a valuable business asset, especially for many companies in the area of online marketing and advertising. The value of personal information is reflected in financial success of the companies that offer free-of-charge, high quality products in exchange of using its subscribers private information, with Facebook being a prominent example. A natural question that appears is: what happens to all the collected data? The popular answer is that this information is used for targeted advertising. It is used to bin the users into specific marketing profiles (so called "personas") according to their needs, interests or preferences, so later the companies can tailor their product offers, prepare personalised advertisements or modify search results accordingly. This practice motivated the research community to create tools to uncover correlations between personal information and the delivered advertisements [50]. Moreover, search engines can use personal information to personalize search results. This personalization can lead to *filter bubble* effect [68], where the user is separated from the information that does not match his profile, and in extreme case he is unable to access a particular information at all. Hannak *et al.* in [39] tries to quantify this elusive phenomena, and presents empirical evidences of the filter bubble effect. Also, Xing *et al.* in [85] presents a tool that enables Internet users to examine if they are subject to the filter bubble effect.

In our work we evaluate different hypothesis, namely that this information is used for price discrimination. In particular, this information can be used to estimate the user's willingness to buy a particular product, reflected by the *reservation price* – the maximum price that the customer is willing to pay for the product. A retailer that is able to estimate consumer's reservation price, can alter the price of the same good offered to different users. This practice, where exactly the same product is offered to different users with different prices, is known as *price discrimination*.

The economic phenomenon of price discrimination (PD) existed before the Internet, but as the Internet enabled new ways of circulation of the information, it also enabled new ways to price-discriminate [65]. It is considered desirable by economists as positively affecting the effectiveness of the markets [64]. On the other hand, PD is not well received by customers. For instance, in 2000 Amazon was heavily criticised by the online community when it turned out that it showed different prices for regular and accidental customers [27]. Price discrimination practices in the area of academic journals are well known [66]. "Personalized pricing" was even a subject of a patent filling by eBay [9]. Also, in 2012 Orbitz, a large online travel agency, was criticised when they were found to present different offers to regular users, and to Mac users. Although the second case is rather *search discrimination* than price discrimination, both cases show that the online consumer community is very sensitive to every symptom of using their personal data to alter the offers and prices.

Different information vectors can be leveraged to facilitate price discrimination. For instance, a user buying luxury products frequently, using a more expensive computer to access the retail website, or whose Internet connection can be mapped to a ZIP code associated with a reach neighbourhood, might be willing to pay more than an average user. We empirically analyse those information vectors (technological, geographical, and personal information) in Chapter 4. Although we find empirical evidences of PD, we also conclude that analysing hand-picked websites does not allow to scale such experiment effectively. Such methodology is limited by the researcher's capability to run the experiments, and does not allow to explore information vectors that are were not thought beforehand. Also, this method does not allow a regular Internet user to examine if he is not subject to price discrimination. To this end, in Chapter 5 we show that *crowd sourcing* is a feasible method to investigate price discrimination. We build and deploy a system that allows a regular user to compare an arbitrary price in the Internet from different geographical locations. Existence of PD was reconfirmed in [40], where the authors conduct a thorough study of personalized pricing and price steering. The study shows inconsistencies in prices of products and services of some of the top car rentals, hotel booking services and online retailers, presented to a control group of accounts and to the real users. As more of the everyday activities move to the Internet, circulation of the personal information and its impact on the Internet economy becomes even more important research area.

1.2 Thesis organization and contributions

This thesis is divided into two parts, shedding light on two areas related to measurements in the economy of the Internet.

Part I is devoted to the macroeconomic aspects of the Internet economy, namely to characterizing and synthesizing the interdomain traffic matrix. In Chapter 2 we analyse the interdomain traffic from a European-wide network and characterize some important spatial properties of the ITM. We confirm previous findings about sparsity and low effective rank of the traffic matrix. We find that traffic sourced by AS-es is heavy-tailed, and that the statistical distribution of the traffic can be modeled as either Pareto or LogNormal. We find some evidence of relation between the shape of the traffic and congestion within the network. We also find significant correlations between the rows in the ITM, which results from a high popularity of a small set of prefixes. Later in Chapter 3 we analyse and model the interdomain traffic at the level of connections and take into account the relative sizes of the ASes. We model multiple application types by manipulating forward- and reverse traffic ratios that the particular application produces. Moreover, we capture differences in regional popularity of different content. Eventually we present a tool to synthesize synthetic traffic matrices.

Part II focuses on microeconomic aspects of the Internet economy, that is on price and search discrimination. In Chapter 4 we empirically demonstrate the existence of signs of both price and search discrimination, and we analyse the information vectors used to facilitate them. In particular, we find evidence of price differentiation based on geographical location, and based on the originating URL (i.e., URL of a page that redirected to a particular product). We also find signs of search differentiation based on personas' traits. Next in Chapter 5 we present a crowd-sourcing study on price differentiation, using a distributed system called $heriff, especially built for that purpose. We show that crowd-sourcing is a feasible way to find instances of price differentiation, and we analyse particular instances of PD found using this system. We show a connection between different location and pricing, and also that the customer profile can impact product price.

Working with network measurements implies processing large quantities of on-line data. Findings that result from our initial exercise in this area are presented in Appendix A. Although the work presented there does not concentrate on Internet economy *per se*, the methodology developed during this exercise was later used to effectively process large volumes of traffic data.

1.2.1 Publications and other activities

The following papers were published as an outcome of the research presented in this thesis:

- *"Towards a statistical characterization of the interdomain traffic matrix."* Jakub Mikians, Amogh Dhamdhere, Constantine Dovrolis, Pere Barlet-Ros, and Josep Solé-Pareta. IFIP Networking conference, Prague 2012.
- *"ITMgen - A first-principles approach to generating synthetic interdomain traffic matrices."* Jakub Mikians, Nikolaos Laoutaris, Amogh Dhamdhere, and Pere Barlet-Ros. IEEE International Conference on Communications – ICC, Budapest 2013.
- *"Detecting price and search discrimination on the Internet."* Jakub Mikians, László Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. The Workshop on Hot Topics in Networks – ACM HotNets, Redmond 2012.

- *"Crowd-assisted search for price discrimination in e-commerce: first results."* Jakub Mikians, László Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. International Conference on emerging Networking EXperiments and Technologies – ACM CoNEXT, Santa Barbara 2013.

- *"A practical approach to portscan detection in very high-speed links."* Jakub Mikians, Pere Barlet-Ros, Josep Sanjuas-Cuxart, and Josep Solé-Pareta. Passive and Active Measurement – PAM, Atlanta 2011.

The other activities related to the research presented in this document:

- Our work on price discrimination on the Internet, presented in Chapter 4, was mentioned in several The Wall Street Journal articles [14, 15, 17].

- Research described in Part II was presented at LAP/CPC/ICPEN conference (Antwerp, 16-17 April 2013) as invited talk. London Action Plan (LAP) is a network of anti-spam government authorities and leading technologists that shares investigative intelligence, coordinates law enforcement, and develops training to address spam and other cyber threats through civil and administrative enforcement. International Consumer Protection Enforcement Network (ICPEN) and EU Consumer Protection Cooperation Network (CPC) are focused on broad enforcement and policy consumer protection initiatives.

- I am co-author of a paper on personalized advertising: *"Understanding Interest-based Behavioural Targeted Advertising"* Juan Miguel Carrascosa, Jakub Mikians, Ruben Cuevas, Vijay Erramilli and Nikolaos Laoutaris, CoRR, 2014.

- I interned twice at Telefonica Research in Barcelona in 2012 and 2013. During the three month stays I worked on price discrimination and the results presented in Part II are the direct outcome of those internships.

- In 2011 I conducted a two month stay at Georgia Tech, working on characterization of the Interdomain Traffic Matrix. Research described in Chapter 2 is the outcome of this stay.

Part I: Macroscopic view – analysing and synthesizing the Interdomain Traffic Matrix

Chapter 2: Analysis of the Interdomain Traffic Matrix

The knowledge of interdomain traffic characteristics is important for a number of reasons, particularly related to economics and policy, as the flow of money on the Internet typically follows the flow of traffic. Even though interdomain traffic patterns significantly impact the evolution of interdomain topology and economics, Internet pricing, and policy considerations (e.g., network neutrality), we have little knowledge of the global Internet Interdomain Traffic Matrix (ITM) and of its dynamics. The major obstacle to infer interdomain traffic characteristics has been lack of data, at least in the research community. As such, accurately measuring the complete ITM is likely to remain an elusive goal. Even if direct measurements of the ITM are unlikely to be available, there is value in measuring qualitative properties of the ITM that can then be used to better inform Internet economics and policy research.

In this chapter we infer some statistical properties of the interdomain traffic matrix. We rely on passive flow data from the GÉANT network, the largest academic/research backbone in Europe that connects hundreds of universities and research organizations to the global Internet. Using this data, we directly measure the ITM elements that are routed via the GÉANT network. *We emphasize that our goal is not to accurately measure each entry of the ITM.* Instead, we aim to *infer statistical properties of the ITM from the elements that we can observe at GÉANT.* We believe that such properties of the ITM can yield a better understanding of its nature and can be used to generate synthetic, but realistic ITMs for simulation and modeling purposes. We are aware of the limitations of the analysed dataset: GÉANT, as a European academic network, it is not representative of the whole

Internet. Nevertheless, it is one of the most complete datasets of interdomain traffic available to the research community, and we hope that the findings presented here will serve for a better understanding of interdomain traffic.

We focus on *spatial* properties of the ITM. In particular, we characterize the visible portion of the AS-to-prefix traffic matrix. We confirm previous results about the sparsity and low effective rank of the ITM. We find that the distribution of traffic sourced by ASes is heavy-tailed, but the exact nature of the distribution can be between Pareto and LogNormal, depending on the source AS. We conjecture that the exact shape of the distribution could be related to congestion within the source AS. We also find significant correlations across different rows of the ITM, mostly due to relatively few highly popular prefixes.

2.1 Datasets

2.1.1 Traffic data

Our approach relies on using traffic data collected from a "network in the middle", i.e., a network that provides transit services to edge networks. To this end, we use traffic data from the GÉANT network [79], a Europe-wide backbone provider spanning 34 countries and connecting over 30 million researchers and students, with an overall throughput of about 50 Gb/s. GÉANT customers are mainly universities and national research networks; consequently, the traffic at GÉANT does carry an academic bias. Nevertheless, approximately half of the traffic is directed to commercial networks. For most of the connected entities, GÉANT is not the only network provider, so only a part of their traffic can be observed. Also, ASes connected to GÉANT are usually not stub networks, but can contain many subnetworks, e.g., National Research and Education Networks (NRENs) connecting many national universities. In the rest of this chapter, all ASes for which we analyse traffic are research and academic networks that GÉANT is serving.

We collect NetFlow traffic summaries from 18 routers at GÉANT points of presence (POPs) for all traffic entering the GÉANT network. As GÉANT is a transit network and the traffic is neither locally produced nor consumed, we measure all traffic entering and leaving the network by combining the information from the 18 POPs. Because the GÉANT NetFlow data is sampled at the rate of 1/100, we estimate bytes and packets by dividing them by the sampling rate[1]. We determine the source and destination ASes by mapping the source and destination IP addresses from NetFlow records to the corresponding ASes. Previous work defined an ITM at the AS-to-AS granularity [26, 23], i.e., ITM element $T_{i,j}$ measures the traffic sent by a source AS i to destination AS j. However, as ASes do not necessarily route all their traffic through GÉANT, we do not observe traffic to all prefixes originated by the same destination AS. An AS-to-AS ITM would underestimate the traffic to such destination ASes. Consequently, we work with an AS-to-prefix ITM, i.e., we characterize the visible traffic sent from a source AS to each destination prefix over a certain aggregation interval, where a row of the matrix indicates the traffic *produced* by an AS, and a column indicates the traffic *consumed* by the prefix. In the rest of the chapter we will concentrate mostly on the *rows*, as characterizing ASes (rather than prefixes) is more relevant in the context of Internet economics. Table 2.1 describes our traffic data. For `trace W` we observe traffic for about 8×10^6 ITM elements, that is only about 0.06% of the total number of elements in the AS-to-prefix matrix. During that week, the matrix consisted of 36k rows (ASes) and 349k columns (prefixes).

Working with an AS-to-prefix definition of the ITM, we can classify ITM elements into three groups. *Unknown* elements are those that we do not observe in the NetFlow data, as the routing path $i \to j$ does not cross GÉANT. *Visible non-zero* elements are the ITM elements for which we observe some traffic, so $\mathrm{TM}_{i,j} > 0$. Finally, we have *visible zeros*, the elements $\mathrm{TM}_{i,j} = 0$ for which the routing path $i \to j$ crosses GÉANT, but they see no traffic in the aggregation interval over which the ITM is constructed. In Section 2.2.1, we describe how we identify visible elements.

We also collect NetFlow data from the UPC[2] access link. We see all traffic from UPC in that data because this is the only access link at UPC. We use UPC data to validate the sparsity results in Section 2.2.1.

[1] We do not estimate the number of flows, because packet sampling does not sample flows uniformly.

[2] Universitat Politècnica de Catalunya, BarcelonaTech

	`trace W`	`trace M`	`trace Y`
period	1 week Nov 22–28, 2010	1 month Nov 1–30, 2010	52 weeks from Jan 4, 2010
flows	3.91×10^9	1.99×10^{10}	2.17×10^{11}
packets	3.61×10^{12}	1.74×10^{13}	1.70×10^{14}
bytes	3.26×10^{15}	1.55×10^{16}	1.45×10^{17}
NetFlow data volume	111 GB	476 GB	5.75 TB

Table 2.1: Parameters of the GÉANT NetFlow traces.

2.1.2 Routing stability and snapshot length

As described in Section 2.1.1, the ITM is estimated over a certain time interval. If the interdomain routing is stable during that interval, we can be certain that if we observed some traffic for an element $T_{i,j}$, then this reflects *all* traffic sent from i to j in that time interval. If, however, routing is not stable, then $TM_{i,j}$ may reflect only a portion of the traffic sent from i to j during this interval. We need to find an appropriate aggregation period that, on one hand, catches a significant volume of the traffic, and, on the other hand, is affected by routing instability as little as possible.

To examine routing stability, we use BGP data from RouteViews [80] collectors that peer with several hundred ASes to collect BGP tables and updates. We analyzed BGP table dumps from 4 collectors over one month. We are interested only in the routes that cross GÉANT, and so we extracted 9000 AS-to-prefix paths, each of which crossed GÉANT[3] at least once in that month. For each path we examined if it is *stable*, i.e., if it is routed via GÉANT in all BGP snapshots. Note that a path may be seen by one BGP collector as crossing GÉANT, but not crossing GÉANT by another collector.

We define routing stability ρ as the probability that a path through GÉANT does not change during a specified time interval. We find that for a day $\rho = 0.999$, for a week $\rho = 0.952$, and for a month $\rho = 0.750$. We conclude that an aggregation interval of one week provides a good trade-off between the volume of traffic captured by the ITM snapshot and route stability.

[3]GÉANT's AS number appears in the AS path.

2.2 Properties of the ITM

In this section we examine the statistical properties of the measured ITM, particularly sparsity (Section 2.2.1), statistical distribution of ITM rows (Section 2.2.2), and possible causes for the differences across distributions for different source ASes (Section 2.2.3 and 2.2.4).

2.2.1 Sparsity

For a given ITM snapshot, we estimate the sparsity S as the ratio of the number of visible zeros (defined in Section 2.1.1) to the number of all visible elements. In the case of our data this is problematic, since we cannot directly distinguish visible zeros from unknown elements. We next describe an approach to estimate a *lower bound* on the sparsity of the AS-to-prefix ITM.

Assume, initially, that the routing path between source i and destination prefix j is stable. Let T refer to the AS-to-prefix ITM measured over a certain aggregation interval, for which we estimate the sparsity. Let R be another instance of the AS-to-prefix ITM, aggregated over a larger time interval. We refer to R as a *reference ITM*. If $T_{i,j} = 0$ and the same element $R_{i,j} > 0$ then $T_{i,j}$ is a visible zero - we are sure that $i \to j$ is routed via GÉANT (because we saw some traffic in the reference ITM). If the aggregation interval for snapshot R is larger than (and overlaps with) that of T, we can identify *some* of the visible zeros in T. Let n_R be the number of visible non-zeros in R, and n_T the number of visible non-zeros in T. Then $n_0 = n_R - n_T$ is the number identified visible zeros in T. The lower bound of the sparsity of T is then $S = n_0/n_R$. This is a lower bound, because not all visible non-zeros in T can be identified (we cannot identify the elements that are visible zeros both in R and T).

The longer the aggregation interval for R, the more visible zeros in T we can identify. However, the longer the aggregation interval, the lower the routing stability ρ (see Sec. 2.1.2). If path $i \to j$ is not stable, then we could see that $R_{i,j} > 0$ and $T_{i,j} = 0$, but the cause is that this path was routed via GÉANT for R and not routed via GÉANT for T. The real number of visible zero elements in T is lower bounded by $\rho(n_R - n_T)$. Therefore, the lower bound of the sparsity is

$S = \rho(n_R - n_T)/n_R$.

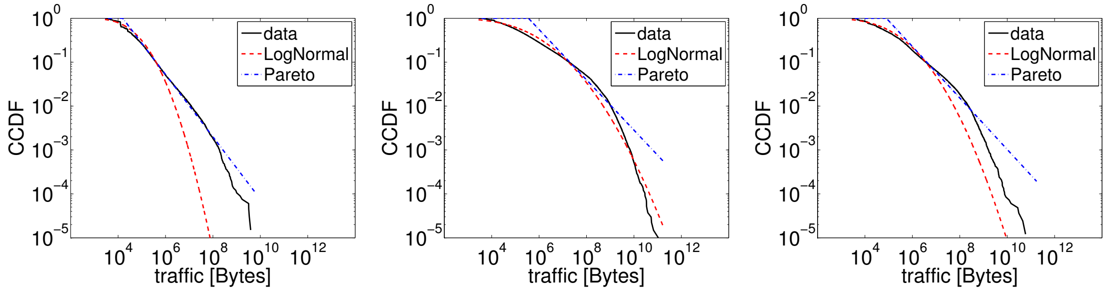
We estimate the sparsity for ITM snapshots aggregated over each week in `trace Y` and over each day in `trace W`. In the former, we constructed the reference snapshot by aggregating over one month, while in the latter the reference snapshot was over one week. The average estimated lower bound of the sparsity for the weekly snapshots in `trace Y` is 0.26, which means that *at least* 26% of the ITM elements are always zero. For the daily snapshots in `trace M`, the lower bound of the sparsity is 0.47. We also observed weekly trends in the sparsity – the estimated sparsity of the daily ITM is higher during weekends (we omit the graphs due to space constraints).

We also examined the traffic measured at the UPC access link, which is equivalent to observing one fully visible ITM row. For a single week, we observed no traffic to 45% of the destination prefixes, i.e., 45% of elements in this row were visible zeros. The results we report here corroborate the observations by Gadkari et al. [36]. Those authors observed that for the traffic sent from a regional ISP, during a single day, 49% of the destination prefixes were not used.

2.2.2 Distribution of traffic generated from each AS

Heavy-tailed distributions are commonly observed in the Internet [25, 30, 20]. It is not surprising that we also see heavy-tailed distributions for the generated traffic from each AS in the AS-to-prefix ITM. We analysed the distribution of generated traffic in ITM snapshots for each week in `trace Y`, selecting only those ASes (rows) for which traffic to a significant number of prefixes is routed via GÉANT (we set this threshold to 10k prefixes). In total, we analyze 3189 rows (119 distinct ASes in all 52 weeks). We find evidence for heavy-tailed distributions in the majority of the rows (94%) – the top 15% of the destination prefixes account for over 95% of the traffic. For the remaining 6% of the rows, the top 15% of prefixes account for over 71% of the traffic. In the remainder of the chapter, we refer to the "tail of the traffic distribution" as the traffic sent to the top 15% of destination prefixes by the corresponding AS.

Figure 2.1 shows the distribution of the traffic generated by three ASes, as an example. The tail of the distribution in Figure 2.1a can be modeled as Pareto



(a) Pareto-like ($D = 0.88$) (b) LogNormal-like ($D = 0.27$) (c) In the middle ($D = 0.43$)

Figure 2.1: Instances of the generated traffic distribution. The tail of the distribution varies between the “straight” Pareto-like to the “bent” LogNormal-like.

(the CCDF in log-log scale resembles a straight line), while the distribution in Figure 2.1b can be modeled as LogNormal. This confirms previous observations of the heavy-tailed nature of sourced traffic distributions [20, 32, 63] with a more recent dataset. On the other hand, the distribution in Figure 2.1c decays faster than Pareto but slower than LogNormal. The values in the tail refer to “heavy” prefixes, i.e., destinations that receive the largest fractions of traffic. The tail of LogNormal decays faster than the tail of Pareto, and so there is a higher probability of observing heavy destination prefixes at source ASes that follow the Pareto distribution than the LogNormal. We analyze a potential cause for this difference in the distribution shape in Sec. 2.2.4.

We next describe a method to determine whether the distribution of ITM elements for a row follows the LogNormal or Pareto distributions. We could use the Kolmogorov-Smirnov (K-S) or other goodness-of-fit tests. However, we are mainly interested in characterizing the *tail of these distributions*, ignoring the values in the main body of the distribution. This is because, due to NetFlow sampling, the body of the distribution consists of small values that are noisy.

Let X be the examined sample and F be the empirical distribution of X. The tail of X consists of all values in the top 15-percentile of the distribution, i.e., the values above some “tail threshold” τ. Let F' be a candidate distribution (LogNormal or Pareto) that we try to fit to the tail of X. From the candidate distribution F' we generate a sample X'. We then generate a sample $\hat{X}$ by combining the tail of X and the body of X'.

$$\hat{X} = \{X' : X' < \tau, X : X \geq \tau\}$$

We now apply the K-S test under the null hypothesis H_0 that $\hat{X}$ is drawn from the same distribution F'. By construction, both $\hat{X}$ and X' have the same bodies and they differ only in their tails. Therefore, the differences between $\hat{X}$ and X' reported by the K-S test should be caused by the differences in the tails. If H_0 is rejected for a LogNormal candidate distribution and not rejected for Pareto, we assume that the tail of the data fits Pareto. In the opposite case the tail is modeled as LogNormal.

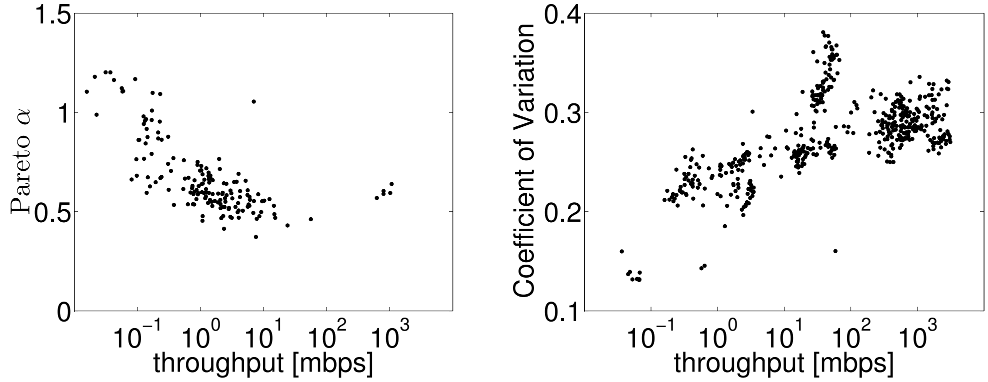
We applied this method on the traffic distributions of 3189 ASes, of which 504 were classified as LogNormal and 162 as Pareto. Our method does not classify the majority of ASes as either Pareto or LogNormal. In those cases, the empirical distribution seems to be between the previous two models.

2.2.3 Distribution parameters

To generate synthetic distributions of sourced traffic, we need to know the nature of the distribution (Pareto or LogNormal) and the associated parameters. In particular, we are interested in the "shape" parameter of these two distributions.

We investigated whether the shape of the measured distributions depends on the AS traffic throughput, i.e., on the total traffic generated by that AS. The shape of the Pareto distribution is represented by the α parameter; lower values of α indicate a heavier tail. For LogNormal, we characterize the shape of the distribution using the coefficient of variation (CoV); a higher CoV indicates a heavier tail.

Figure 2.2a shows α and Figure 2.2b shows CoV as a function of the average AS throughput. Clearly, in both cases, an increasing throughput causes a change in the shape parameter – the tail becomes heavier, and the more popular destinations receive even more traffic. This is not obvious, because the increasing traffic could cause only changes in the *scale*, but not necessarily in the *shape* of the distribution. The values of the Pareto α parameter are between 0.37 – 1.20, while the LogNormal CoV varies between 0.13 – 0.38.



(a) Pareto α (b) LogNormal, coefficient of variation

Figure 2.2: Distribution parameters as a function of throughput.

2.2.4 What determines the shape of the tail?

In this section we investigate why the generated traffic distribution follows a LogNormal tail for some ASes, and a Pareto for others. We also show that this difference could be related to congestion within the corresponding AS.

Shape and throughput

To compare the shape of the previous distribution, we define a metric D that indicates if the tail is LogNormal-like or Pareto-like. Let F be an empirical CDF of the sample, and let F_P and F_L be the CDFs of the Pareto and LogNormal distributions that fit the tail of the sample. We measure the difference in the tail using the Kolmogorov-Smirnov metric: $KS(F_1, F_2) = max|F_1(x) - F_2(x)|$ *only for values of* x *that are in the tail.* We define D as

$$D = \frac{KS(F, F_L)}{KS(F, F_L) + KS(F, F_P)} \tag{2.1}$$

where $D = 0$ indicates that the tail follows a LogNormal distribution, $D = 1$ indicates a Pareto distribution, and values in between represent how close the sample is to each of those two distributions.

In Figure 2.3 we plot the metric D and the overall throughput for each examined AS in a single week (`trace W`). The dot size indicates the number of visible-non

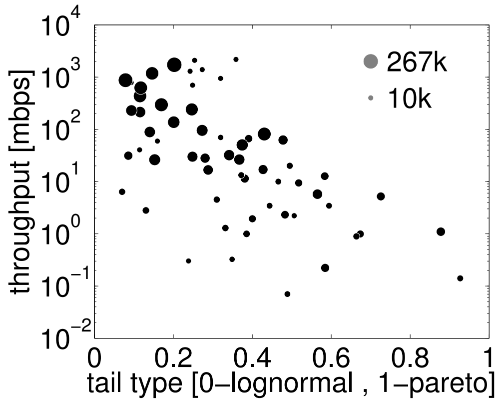


Figure 2.3: Type of the distribution tail and average throughput. Each dot is a separate AS. The dot size indicates the number of visible non-zero prefixes.

zero prefixes for that AS. Visually, we see that ASes with lower throughput are more Pareto-like and ASes with larger throughput have a more LogNormal-like tail.

The reader may be concerned that the relation seen in Fig. 2.3 is an artifact of visibilty – the fact that we do not observe traffic from each source AS to the same set of destination prefixes. We investigated this possibility by performing the following experiment. Let AS_P be an AS with Pareto-like distribution and let AS_L be an AS with LogNormal distribution. Let Q be the set of prefixes that are visible non-zeros for *both* AS_P and AS_L. We determine whether the traffic sent from AS_L to prefixes Q follows the distribution of AS_P or AS_L. If it follows the distribution of AS_L, then it means that the distribution does not depend on the number of observed prefixes. We selected 4 Pareto-like ASes (with between 19k and 57k visible non-zero prefixes) and 10 LogNormal-like ASes (with between 120k and 260k visible non-zeros) and examined all 40 pairwise combinations. Interestingly, in all cases the distribution of the traffic sent by AS_L to prefixes in Q retained the properties of AS_L. We thus reject the possibility that the shape of the generated traffic distribution is a function of the number of observed prefixes.

Congestion

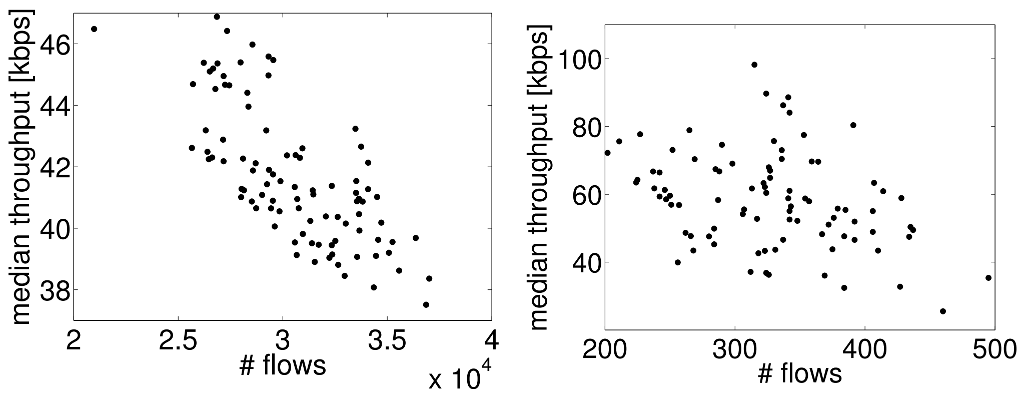
In this section, we investigate a possible reason why some ASes follow the LogNormal distribution and others the Pareto distribution. Cha et al. [25] show that

Pareto "tail truncation" effect can be caused by bottlenecks. In the case of interdomain traffic, we suppose that tail truncation is caused by bandwidth bottlenecks. Specifically, we conjecture that congestion can "push" the generated traffic distribution from the Pareto distribution towards the LogNormal distribution. It would mean that congestion affects large ASes more than the small ones. Finding evidence, and explanation, of congestion inside networks is a challenging task, as we do not have any direct information about the ASes connected to GÉANT. We only have NetFlow data *collected at GÉANT* for a subset of destination prefixes; we plan to confirm these observations with more exact traffic samples as part of future work.

To detect congestion, we follow the intuition that during periods of congestion, every additional connection at the link will compete for throughput with existing connections. Consequently, we should see a negative correlation between the number of active connections at a link and the median throughout of each connection. We analyzed NetFlow data for two ASes (one LogNormal-like and the other Pareto-like) over three days at the time period that congestion is most likely (10:00–20:00), with bins of 20 minutes. To reliably estimate flow throughput and to discard TCP control flows, we only consider flows with at least 5 sampled packets, at least 100B each. Figure 2.4 shows the number of flows and the median throughput per flow for both ASes. For both ASes, we measured the Spearman correlation coefficient for each day. For the LogNormal-like AS, the daily correlations are -0.85, -0.77 and -0.82. For the Pareto-like AS we do not see any significant correlation. In summary, there is some evidence that ASes with LogNormal traffic distribution are subject to congestion, at least for certain time periods, while ASes that follow the Pareto distribution are not subject to congestion.

2.3 AS correlations and popular prefixes

In this section we show that the ITM rows are not independent. For example, this can be the case when a set of destinations is popular for several source ASes. Correlations across rows are important in matrix completion techniques that attempt to estimate unknown elements in one row using known values in other rows. The correlations are also useful for generating synthetic ITMs.



(a) Congested (b) Not congested

Figure 2.4: Number of flows and the median throughput for a LogNormal-like (a) and Pareto-like (b) AS. 22–24 Nov 2010, 10:00–20:00. A few extreme outliers in (b) are not drawn.

2.3.1 Correlations

The nature of our dataset makes it challenging to directly measure correlations between rows, as two ITM rows can observe different sets of destination prefixes. Even if we could observe two complete ITM rows, we should not expect to see very high correlation between them, as each row consists of only few large values, with the bulk of the distribution consisting of small and highly noisy values. Hence, we restrict ourselves to studying correlations only for the set of heaviest prefixes in each row. To measure correlations between two rows of the ITM, we retain the top 15% of prefixes in each row, and calculate the Spearman correlation across prefixes that are present in both rows. We calculate pairwise correlations in this manner for each pair of rows in `trace W`. To obtain more accurate results, we only consider rows with at least 3000 visible non-zero elements. To calculate the correlation between two rows, we require that the overlap between them is at least 100 prefixes.

Using this method, we measure the correlations between 15146 pairs of rows. 10931 pairs give statistically significant correlations ($p<0.01$). 99% of the correlations are positive; the average correlation is 0.28. The highest correlation is 0.85 and 408 pairs of rows have a correlation larger than 0.5. Interestingly, for 135 pairs of rows with an overlap of more than 10000 prefixes, we observe an average

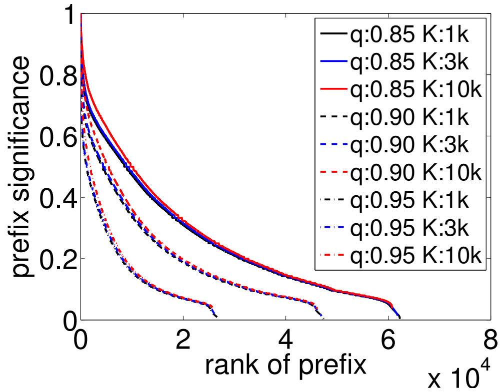


Figure 2.5: Significance of prefixes, ordered. Only prefixes observed in at least 20 rows are considered.

correlation of 0.44.

2.3.2 Popular prefixes

The previous section raises the question of whether there are some globally "significant" (or popular) prefixes, i.e., prefixes that account for a large fraction of the total traffic generated by each AS. We define a prefix p as *significant* for source AS i if p is in the top-q quantile of the visible non-zero elements in row i. If $n(p)$ is the number of ASes that send traffic to p via GÉANT and $n_S(p)$ is the number of ASes for which p is significant, then the *significance* of p is $I(p) = n_S(p)/n(p)$. For the sake of accuracy, we consider only rows with at least K visible non-zero elements, and prefixes with $n(p) > 20$. We experiment with different values of K and q. Figure 2.5 shows, for each prefix p the significance value $I(p)$, for different values of K and q. The curves for different values of K and q are similar, at least in shape. Interestingly, there are some prefixes that are significant for most ASes (I values close to 1). For instance, for K=3000 and q=0.85, 460 out of 61000 prefixes have significance value of 0.8 or higher, and those very popular prefixes receive on average 32% of the total traffic produced by the corresponding ASes. 8800 prefixes with $I(p) > 0.5$, account for about 78% of the traffic.

This implies that there is a small group of prefixes which are significant for almost all source ASes. We found by manual inspection that more than 25% of these very popular prefixes ($I(p) > 0.8$) belong to well known large Internet entities

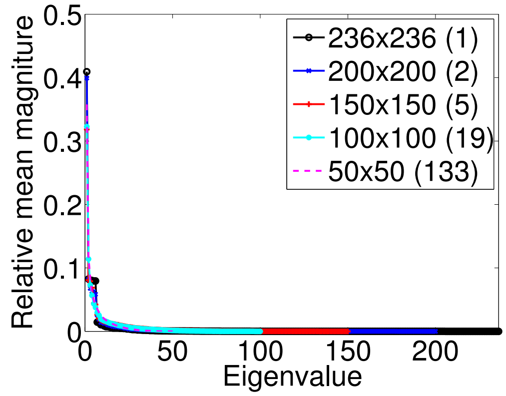


Figure 2.6: Eigenvalues of the submatrices (relative magnitudes). Only a small number of the values is significant, what indicates a low effective rank.

(such as Google, NTL Virgin, OVH, Level 3, to name a few).

2.3.3 Low effective rank

A matrix that has *low effective rank* can be approximated by a linear combination of a small number of independent rows or columns. Some techniques to estimate invisible elements of the ITM (e.g., matrix completion [23, 87]) rely on the fact that the ITM has low effective rank.

To study whether the AS-to-prefix ITM has a low effective rank, we used an ITM snapshot from `trace W`, identifying visible zeroes using a monthly reference snapshot (see Sec. 2.2.1). To examine the rank of the matrix, we adapted the methodology used in [23]. From the observed ITM we extracted square visible submatrices of various sizes, and calculated the eigenvalues for these submatrices. Figure 2.6 shows the normalized (sum to 1) and averaged eigenvalues across the extracted submatrices. Clearly, only about 10 eigenvalues are significant (even for the submatrix that is 236-by-236 elements) , meaning that the ITM can be approximated with a relatively small number of independent vectors. This observation remains independent of the size of the submatrix, indicating that the global ITM is also likely to be of low rank.

2.4 Related Work

Given the importance of characterizing interdomain traffic demands, there has been surprisingly little prior work on estimating the characteristics of the interdomain traffic matrix. The major reason for this, unfortunately, has been the lack of publicly available data [79] to enable such a study by the research community. An early study by Fang et al. [33] showed that interdomain traffic distributions are highly non-uniform, an observation that has since been confirmed by others [20, 63]. Feldmann et al. [35] described a method to estimate web traffic demands using data from server logs at a large content delivery network. Chang et al. [26] propose a method to estimate interdomain traffic demand by estimating the importance of an AS in various roles – residential access, business access and web hosting. In contrast, our work aims to extract relevant statistical properties of the ITM from direct measurements, which resembles the approach in [63] for intradomain traffic. Sen et al. [74] analysed P2P traffic in large networks. A recent study from Arbor networks [49] revealed some important characteristics of interdomain traffic, such as the increasing dominance of large content providers. That study does not, however, measure a traffic matrix. Gadkari et al. [36] study prefix activity from a source AS, discovering that only a small fraction of destination prefixes receive traffic during a day, indicating that the ITM is sparse. Bharti et al. [23] also report on the sparseness of the ITM, and propose methods to infer the invisible elements of the ITM. Our work confirms the sparsity and low effective rank of the ITM seen in previous work [87].

2.5 Conclusions

In this chapter we analysed selected statistical properties of the Interdomain Traffic Matrix using data from a large research network. First we investigated stability of the routing in the network and found optimal aggregation intervals that allow us to capture significant portions of traffic without being affected by changes in the routing in GÉANT. After that we investigated sparsity of ITM. We determined that the lower bound of the sparsity is 47% for the weekly snapshots of the traffic, and 26% for daily snapshots. Next we investigated shape of the statistical distributions of the traffic generated by the ASes and we found that they are predominantly heavy tailed. Among those distributions, we found examples of both LogNormal and Pareto. We observed that the parameters of the distributions vary with the throughput of the traffic, and that one possible explanation could be congestion in the networks sourcing the traffic. Next we examined correlations between rows of the matrix and found that the rows are highly correlated. This high correlation is naturally related with high effective rank of the matrix, which we investigated afterwards.

Chapter 3: Synthesization of Interdomain Traffic Matrices

In the previous chapter we focused on inferring statistical properties of the interdomain traffic matrix. This chapter focuses on generating synthetic interdomain traffic matrices. We need realistic interdomain traffic matrices in order to model and simulate new interdomain interconnection policies, pricing schemes, or routing protocols. Moreover, simulations of the interdomain Internet often need to be at different scales than the real Internet (which consists of more than 40,000 networks), either "shrinking" the actual traffic matrix for scalable modeling and simulation, or to investigate "what-if" scenarios in the evolution of the Internet. Researchers have mostly had to rely on synthetic interdomain traffic matrices generated using ad-hoc methods, reproducing some high-level characteristics of the interdomain traffic matrices such as heavy-tailed traffic volume distributions, or the presence of large traffic sources and sinks [20, 33, 49]. However, the research community lacks a configurable tool for producing synthetic traffic matrices of arbitrary size that match basic real interdomain traffic characteristics in more detail.

To fill the gap, we present in this chapter the design and evaluation of `ITMgen`, a new tool to generate representative synthetic interdomain traffic matrices. `ITMgen` is based on first-principles, and incorporates several features that result in more representative traffic matrices than the current state of the art [26]. First, we model interdomain traffic at the level of *connections*, taking into account the relative sizes of ASes measured by the number of users they serve. Second, we model multiple content (or application) types, and their effect on interdomain traffic in terms of the ratio of forward to reverse traffic that each application type produces. Third, `ITMgen` captures the fact that the popularity of content objects shows regional effects - certain websites, for instance, may be more popular in specific countries or geographical regions. Finally, `ITMgen` is designed to be parameterized with high-level input data that is available publicly, and we provide such a *canonical* parameterization that represent present-day interdomain traffic characteristics. `ITMgen` is designed to be highly configurable and extensible; when new content types emerge and data about them becomes available, `ITMgen` can be easily extended to incorporate the new data.

The remainder of this chapter is organized as follows. Sec. 3.5 discusses related work. Sec. 3.1 describes the design of `ITMgen`. Sec. 3.2 describes the datasets used. Sec. 3.3 demonstrates how `ITMgen` can be parametrized and how to synthesize a matrix. The validation is presented in Sec. 3.4. Sec. 3.6 concludes the work.

3.1 ITMgen design

The design of `ITMgen` is based on first-principles, modeling traffic at the level of connections and taking into account traffic asymmetries based on application type and the effects of regional/global content popularity. We emphasize that we focus on generating *static snapshots* of the interdomain traffic matrix. Although such a static model might be sufficient for applications such as Internet economics or network formation, other areas may requite a model that captures temporal effects. We strive to expand the model along the temporal dimension. Next, we summarize the key decisions underlying the design of `ITMgen`.

Connection-based

The interdomain traffic matrix, by definition, is concerned with the terminating ends of the Internet, i.e., it measures the traffic that originates at an AS X and terminates at AS Y. We recognize that such traffic is from *connections that originate from and/or terminate at individual users*. We thus make the design decision to model interdomain traffic at the level of connections, and the traffic exchanged by an AS will depend on the number of users in that AS.

Content types

The Internet caters to a variety of different applications, such as web, peer-to-peer file sharing, streaming video, conferencing, etc. Given a connection, the ratio of traffic flowing in the two directions (traffic asymmetry) over that connection depends on the nature of the application. For example, in the case of client-server applications, the traffic asymmetry will be determined by the ratio of the size of data packets to the size of acknowledgements. In the case of P2P traffic, we expect more symmetric traffic. We explicitly model different application types in `ITMgen`. Note, however, that we are considering different traffic types, and not necessarily network types; the same network can thus host different applications.

Regional and global popularity

We recognize that content popularity on the Internet shows both global and regional effects. With respect to web content, for example, websites such as Google and Facebook are popular worldwide; on the other hand, some websites cater to specific countries or regions. Such regional websites may be highly popular traffic sources for ASes in the same region, but they are not popular for ASes in a different region. `ITMgen` takes into account the global and regional popularity associated with content objects.

Parameterizable using commonly available data

`ITMgen` can be parameterized using commonly available data sources, which measure interdomain traffic characteristics at a high level. Further, we have designed `ITMgen` to be *extensible* to accommodate new application types that may emerge in the future. A user can extend the tool whenever data about new application types - the traffic parameters for the new application type, the global and regional popularities of ASes w.r.t. that application type - are available.

3.1.1 Traffic model

`ITMgen` models the traffic between two ASes as an interaction between users and content within the ASes, facilitated by a set of distinct applications. Consider an example where users in ASes U_1 and U_2 are accessing objects stored on machines in ASes M_1 and M_2, using an application A_1. The volume of this user-to-machine (U2M) traffic depends on the number of users in U_i, the popularity of content in M_j, and the nature of traffic produced by application A_1. Moreover, the popularity of M_1 can be different for U_1 and U_2, for example due to a regional bias. Users in U_1 and U_2 also interact using application A_2, generating user-to-user traffic (U2U). The obvious examples of applications that produce U2M and U2U traffic are browser-based services and P2P, respectively. In our study we omit machine-to-machine (M2M) traffic. The reason is twofold: traffic reports like [8] do not indicate that M2M traffic volume is significant in access networks. Also, access to the packet level data at the level of non-access ASes (*i.e.*, business ASes) is highly restricted. Therefore we only acknowledge that M2M traffic estimation will require further effort.

The traffic represented in the ITM is thus an aggregate of all the individual interactions between users and content in different ASes. There are two levels at which these interactions need to be characterized in order to generate an ITM. At the *macro level*, the traffic between two ASes depends on the number of users and the popularity of the content hosted within those ASes. The common gravity models [26, 73] operate at this level. This level of description is insufficient to capture more elusive aspects of the traffic, namely what happens at the application level. Therefore, we enhance the macro-information with the *micro-level* view which describes the actual interaction between users and content objects.

Combining the macro and micro-level views, traffic from AS i to AS j can be expressed as

$$T_{i,j} = \sum_{\kappa} m_\kappa \left(S_i p_i^\kappa(j) + d_\kappa S_j p_j^\kappa(i) \right) \tag{3.1}$$

S_i denotes the number of users in AS i. $p_i^\kappa(j)$ denotes the relative popularity of j *subjective to i and with respect to application κ.* The two terms in the summation represent the traffic from a user to an object due to application κ, and the traffic produced by that application in the reverse direction. The (a)symmetry in the two directions of traffic due to application κ is denoted by d_κ, and this parameter is application-dependent. The parameter m_κ represents the contribution of each application to the overall traffic mix.

In the rest of this chapter, we describe how to parameterize `ITMgen` with respect to these two applications. Although both groups contain more applications (Skype, mail, etc.), web and P2P in particular contribute to the bulk of interdomain traffic [49, 8]. `ITMgen` can easily be extended to add more application types, as long as the relevant information to parameterize them (m_κ, d_κ and popularity) is available.

A curious reader could notice that we do not consider network topology. Our goal is to model traffic resulting from an interaction between users and content; the user's decision to access a specific content does not depend on the topology.

3.2 Dataset description

We give a brief overview of the datasets we have used to parameterize (Sec. 3.3) and validate (Sec. 3.4) `ITMgen`.

We use the **Alexa** [1] list of global top 1 million websites to measure the popularity of ASes with respect to web content. Alexa also provides per-country statistics[1], which we use to determine the *regional* popularity of ASes. To estimate the popularity of ASes with respect to peer-to-peer traffic, we rely on data obtained by crawling the BitTorrent (BT) tracker (`openbittorrent.com`). To obtain the number of users per AS, we relied on open **marketing reports** from ISPs [2].

To obtain the micro-level information regarding application characteristics (ratio of forward to reverse traffic) and the fraction of traffic accounted for by various application types, we rely on a two-week long **packet level trace from CESCA** [3]. Although CESCA is a fully fledged AS and access to the packet level data at that level is difficult, we strive to confirm our results with other data sources. We deliver `ITMgen` with preconfigured parameters in case a researcher does not have access to the relevant low level data.

For validating `ITMgen`, we use traffic statistics for 3 ISP ASes from **Telefonica**, a world-wide Internet connectivity provider. For those ISPs we analyze traffic statistics for the top 1000 ASes; as the traffic distribution was heavy tailed, those top entries contribute to more than 95% of the total traffic. The statistics come from international access links, therefore some of the regional (country) traffic can be undervalued and we use this data only where this shortcoming is insignificant.

3.3 Parameterization and synthesis

In this section we describe how each of the parameters in (3.1) can be estimated from real-world measurements. We provide this measurement data and the asso-

[1] We use "page views" metric provided by Alexa, together with per-country breakdown.

ciated parameterization as the *canonical parmeterization* of `ITMgen`.

3.3.1 Number of users: S_i

Our model requires an estimate of the number of users in each AS, which we characterize as follows. Using publicly available marketing data and annual reports, we obtained the market shares of ISPs for the top-10 countries in the world according to the number of Internet subscribers [11]. This gives us insight into ISP market shares, but not per-AS estimates. For each ISP, we then obtained the set of ASes belonging to that ISP using *whois* data. For these ASes we measured the number of IP addresses in our BT logs, and split the subscribers of the ISP among different ASes in proportion to the number of IP addresses seen from each of those ASes in BT. The assumption is that approximately the same fraction of users in each AS belonging to an ISP participate in BT file sharing. This gives us an empirical distribution of the number of Internet users per AS, for about 400 ASes. Although this represents only 1% of the total number of ASes, these contribute to about 60% of the total number of Internet subscribers in the world.

In addition to the number of users per AS, we need to determine the fraction of ASes in the world that *do not serve any users*. Such ASes could host content (pure content providers), or provide transit service (pure transit providers). Pure transit providers do not appear in the interdomain traffic matrix, as they do not source or sink any traffic. To find pure content providers, we obtain the set of ASes that host websites represented in the Alexa list. Of these ASes, we separate the ones that do not show any BT activity in our BT logs. We thus find that at least 42% of ASes do not serve end users. We emphasize that these are rough estimates and can be easily improved as more precise data becomes available.

3.3.2 Content Popularity: $p_i^\kappa(j)$

Another macro-level parameter is the popularity of an AS with respect to various content types. Vector $p_i^\kappa(j)$ describes the fraction of traffic generated by an average user in AS i that is sent to AS j. Recall from Sec. 3.1.1 that the traffic between two ASes is proportional to the popularity of the content objects hosted by that AS. Moreover, the popularity can be *subjective*, i.e., some ASes are likely to be popular only in their own region, while others are globally popular. The bias can be easily observed in the rankings of ASes calculated from Alexa: for top-10 most popular ASes in 20 examined regions, by average 53% of them were from that region. As a result, we assign to each AS i a popularity vector $p_i^\kappa(j) : \sum_j p_i^\kappa(j) = 1$ describing the *subjective* popularities of the other ASes, as visible from i.

Web popularity

To gain an insight into the popularity of the WEB traffic, we used *Alexa page views* statistics. Although this metric does not reflect literally the actual traffic volume for the AS, the number of page accesses per AS will impact the generated traffic, and we believe that it can serve as the basis for comparison between ASes. Figure 3.2 shows the distribution of AS popularity for different regions. Strikingly, the underlying distribution appears to be similar for all 20 examined regions (not shown in the figure) and the corresponding Zipf slope falls typically into range $(1.13, 1.28)$. Some ASes, e.g., Google, Facebook, etc. are expected to be popular in many regions. Moreover, an AS that is among the most popular ASes in region A can also be among the most popular ASes in region B. To confirm the intuition, we computed the pairwise Spearman correlation of the rankings in all the considered regions. We found a relatively high Spearman correlation (0.62), indicating that there does exist correlation between the top ranking ASes in different regions.

To synthesize the ITM, we need to define a procedure to create $p_i^\kappa(j)$ that (1) has a certain statistical distribution (resembling measurements), (2) keeps the notion of local and global popularity of ASes (to distinguish between, for example, a global content provider and large regional hosting provider), and (3) preserves ordering (e.g., for two globally popular ASes X and Y, X will be always more popular than Y). The following procedure builds $p_i^\kappa(j)$ for an AS i that captures those three properties. First, we split all ASes into three ordered groups: globally popular, locally popular and the remaining ones. Next, from a Zipf distribution we generate a random vector q (sorted in descending order) of length n, where n is a number of ASes. This vector contains the popularities of remote ASes from the perspective of AS i. Then, n times we pick an AS j from a random group (globally popular, locally popular, or other) and assign the next value from q to $p_i^\kappa(j)$. This way we build $p_i^\kappa(j)$ for a specific AS i.

P2P Popularity

As we mentioned in Sec. 3.1.1 a prevailing U2U application is P2P file sharing, which is responsible for most of U2U traffic between ISPs. In this section we describe the parametrization of P2P popularity vector $p_i^{P2P}(\cdot)$.

We estimate the relative popularity of different ASes for P2P content using BT measurements. To this end, we measured the number of IP addresses from each AS seen in our BT crawls. Figure 3.4 shows the distribution of the active P2P peers per AS. The flat section of the plot suggests an underlying power-law distribution, which is more evident after binning the data. The bent tail could be the effect of an information bottleneck, e.g., insufficient measurement time [25]. To build the popularity vector w.r.t. P2P traffic $p_i^{P2P}(\cdot)$, we draw a vector of random variables from the fitting Zipf distribution with slope 1.63 and assign the generated values, from the highest to the lowest, to ASes in the order of descending number of users. We refrain from modelling regional popularity in P2P traffic, as those effects are difficult to estimate precisely from BT data. As P2P contributes a relatively small fraction of overall Internet traffic [8, 49], we accept the error introduced by not considering the locality of P2P. Nevertheless, it is possible to use measurement-based insights on the regional distribution for P2P [73], together with a procedure similar to the one used for WEB to assign regional P2P popularities.

3.3.3 Application mix: $m_\kappa,\ d_\kappa$

As mentioned in Sec. 3.1, `ITMgen` recognizes the fact that traffic at its micro level is a mix of different applications, which is expressed in (3.1) by κ. There are two crucial parameters we must estimate at the micro-level. The parameter d_κ, which describes the ratio between the two directions of traffic generated by application κ, and m_κ, the fraction of the traffic generated by an average user, due to application κ. These application-specific characteristics cannot be obtained from the macro level data; to this end, we monitored the CESCA access link for 14 days. To classify the applications we used the commercial *PACE* [6] tool for deep packet inspection, which in our case yielded only 13% of unclassified traffic[2].

Our measurements indicate that in the case of WEB traffic the ratio per flow

[2]The overall accuracy was affected by the packet capturing process (e.g., packet drops and truncated flows), which are not related with PACE.

$log_{10}(d_\kappa)$ typically falls into the range $(0.4, 1.5)$ and for P2P traffic into the range $(-0.87, 1.25)$. It is unsurprising that d_{WEB} is skewed, since for WEB one direction of the traffic is predominant. Also the upper bound of d_{WEB} is determined by MTU as $log_{10}\frac{MTU}{TCP\ Ack} \approx 1.56$. Interestingly, the ratio for P2P traffic is both positive and negative, suggesting that some P2P clients use the same connections, once established, to both upload and download the exchanged content. In the latter examples we use the statistical distributions that best fit the measurements, i.e., normal d_{WEB} and uniform and d_{P2P}

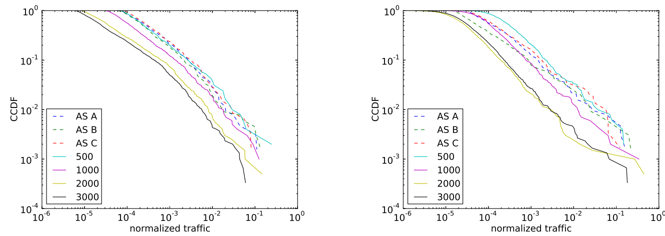
To explain the exact role of m_κ consider the following example: a user downloads a file from a server and $d^{WEB} = \frac{MTU}{Ack}$. Also, the same user exchanges P2P traffic, and $d^{P2P} = 1$. If both applications use the same total bandwidth, it does not mean that the upstream flows (from the user to the object) are the same size: the upstream flow of WEB is smaller than that of P2P. The parameter m_κ reflects this difference in the traffic mix originally generated by an average user. Based on our measurements we choose $m_{P2P} = 0.65$ and $m_{WEB} = 0.35$. We strive to compare those results with the data from other vantage points.

3.4 Validating ITMgen

In this section we validate `ITMgen`. First, we perform some sanity checks to show that a synthetic ITM generated by `ITMgen` reproduces well-known characteristics of the real ITM. Later, we discuss the advantages of `ITMgen` over a common gravity model (GM) [26].

3.4.1 Sanity checks

One of the properties of the ITM observed in [23, 59] is it's low rank, meaning that the matrix can be approximated by a small number of independent vectors. The reason of the low rank is that a small number of the most popular ASes (rows/columns) capture the bulk of the traffic. To verify that `ITMgen` produces ITMs with this property, we computed the eigenvalues of a synthetic ITM with 1,000 ASes. Less than 30 out of 1,000 values were significant, confirming that the low rank property holds for `ITMgen` generated matrices.



(a) Tr. produced (b) Tr. consumed

Figure 3.1: Statistical distribution of the traffic produced and consumed by the observed ASes, for the Telefonica data (dashed line) and the model (solid) for the synthetic ITMs of different sizes.

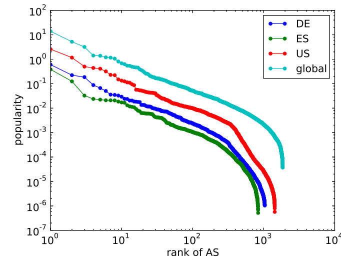
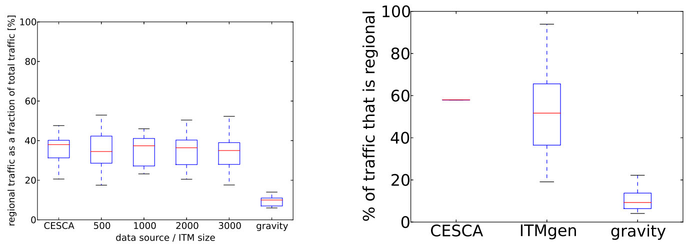


Figure 3.2: WEB popularity distribution of ASes, globally and for three example regions.



(a) Traffic exchanged with ASes within same region; matrices of 4 different sizes are shown.

(b) Regional traffic of CPs.

Figure 3.3: Regional traffic exchange.

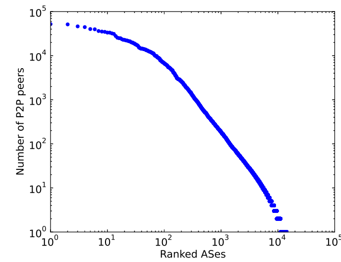


Figure 3.4: P2P activity distribution.

Next, we compare the statistical distributions of the traffic exchanged by ASes in the synthetic ITM and that seen in the Telefonica dataset. Figure 3.1 shows results for 3 ISPs from Telefonica, and selected ASes from the generated ITMs that have a similar number of users, relatively to the total number of users in all ASes. We perform this analysis for synthetic ITMs of different sizes. We observe that the distribution of traffic produced by synthetic ASes is qualitatively similar to that in the measurements. On the other hand, the traffic consumed by synthetic ASes appears to be more skewed than in the measurement data (the slope of 0.89 for the measurements and 0.69 for the model). Although the mismatch is visible, we do not aim to match exactly the special case visible in the plot. Simulation parameters, in particular content popularity distribution, can be adjusted to match desired special cases.

3.4.2 Regional effects

`ITMgen` explicitly models regional biases in content popularity. We present those effects introduced by `ITMgen`, and compare them with real measurements, and with a synthetic ITM produced by GM. For this purpose, we model 10 regions with equal number of ASes. As GM does not introduce locality, it will result in a random assignment of ASes to the regions.

We analyze traffic locality for two types of ASes: ISP and CP. From the generated ITM, we select randomly 25 ASes with a similar relative number of users, and calculate the traffic that those ASes exchange with the ASes within the same region. We repeat the same procedure for the ITM generated by GM. We calculate the regional traffic of each institution in CESCA. Figure 3.3a shows the fraction of traffic that is exchanged with ASes in the same region for matrices of different sizes generated by `ITMgen`, a synthetic ITM generated by GM, and measurement data from CESCA. We observe that CESCA traffic is regionally biased - almost 40% of the traffic is exchanged with ASes within the same region. This bias is also reflected in the synthetic matrices produced by `ITMgen`, regardless of their size, as shown in Fig. 3.3a. We also observe that GM produces an ITM with significantly less regional traffic.

Next, in Fig. 3.3b we compare local traffic from the point of view of CPs in the CESCA data. Although we did not have access to a true CP AS, we analyzed the traffic from/to the content servers inside the CESCA AS. The figure shows that in the measurement data from CESCA, about 60% of traffic is local; the synthetic ITM produced by `ITMgen` shows similar fractions, while GM clearly underestimates regional traffic.

3.4.3 Application mix

An important feature of `ITMgen` is that it offers the possibility of modeling the traffic mix in terms of applications. To this end, we show the application mix resulting from `ITMgen`, and later discuss a what-if scenario that considers a new application.

Various reports [49, 8] and our DPI measurements at CESCA suggest that P2P contributes between 9% and 21% of the overall traffic. In the synthetic ITM we observe that P2P contributes an average of 27% of the traffic. The overestimation can stem from the fact that we model only two applications, whereas the mentioned measurements consider all possible applications.

3.4.4 Use case - cloud storage

Here we discuss how to introduce a new application to the ones already modeled by `ITMgen`. We consider "cloud storage" (ST), a service that allows a user to synchronize her data over a cloud that is managed by an external enterprise.

Recall that to model a new application, the user must specify both the macro and micro-level properties of that application. First, we consider the macro level characteristics of ST, expressed by the popularity vector $p_i^{ST}(j)$ (see Sec. 3.3.2). Recall that p describes the popularity of AS j, as seen from AS i. We consider a hypothetical scenario where the storage is provided by three major global content providers, and we assign $p_i^{ST}(j)$ proportionally to $p_i^{WEB}(j)$ so that the more an popular AS already is, the more ST traffic it will attract. Next, we specify the micro level parameters. We simulate that the users generate an additional 5% of upstream traffic due to ST ($m_{ST} = 0.05$). We also simulate that the traffic generated by ST is skewed (upload files from one point and send to many points), and we model the traffic ratio $log_{10}(d_{ST})$ with a normal distribution $N(0.7, 0.2)$.

This information is sufficient to model the new application. We generated synthetic ITMs with `ITMgen`, considering the new application in addition to Web and P2P traffic. Analysis of the synthetic ITMs suggests that ASes providing cloud storage will increase their traffic from 16% to 20%, and the overall traffic generated by all ASes will increase by 9.1%. This example shows how `ITMgen` can be used to model various what-if scenarios related to new application types.

3.5 Related work

Most prior work on traffic matrix estimation and generation focused on *intradomain traffic* (see [23, 20, 63, 71, 86, 87] and references therein). Although those solutions give useful hints about synthesizing interdomain traffic matrices, they cannot be applied directly to the interdomain context. A prior paper on modeling intradomain traffic that inspired our work was by Erramilli et al. [32], which modeled intradomain traffic at the level of individual connections.

Several studies have measured interdomain traffic characteristics. An early study by Fang et al. [33], confirmed by [20, 63], showed that interdomain traffic distributions are highly non-uniform. Labovitz et al. [49] reported that interdomain traffic has been consolidating. Maier et al. [52] characterized residential broadband traffic. Bharti et al. [23] report on the sparseness of the ITM, and propose methods to infer the invisible elements of the ITM. Mikians et al. [59] confirmed the sparseness of ITM, heavy-tailed distribution of sent and received traffic volumes, and measured the global and regional popularity associated with content sources. Feldmann et al. [34] present a methodology to estimate web traffic demands by analyzing CDN logs. While these studies do not directly measure ITM, the research community has mostly relied on measurements reported in these studies to synthesize ITM for modeling and simulation purposes.

The only work presenting a full approach to model interdomain traffic matrices by Chang et al. [26], which uses the *gravity model* to estimate the traffic between the ASes. The authors model ASes with a mix of "utilities" (business, residential, web hosting) and attribute the traffic accordingly to the interacting AS types. In contrast, we do not attribute types or "utilities", but rather distribute users and content, and model their interactions. A further difference is that our model is *topology agnostic*, and does not require knowledge of the interdomain topology in order to synthesize an ITM.

3.6 Conclusions

Modeling the interdomain traffic matrix is a challenging task, as it is impossible to obtain its full view. In this chapter, we present `ITMgen`, a tool to build synthetic

ITMs of arbitrary size. To the best of our knowledge, `ITMgen` is the only alternative to the current state of the art in interdomain traffic matrix estimation [26]. `ITMgen` takes a first-principles approach, and differs from that work in several significant ways - it models traffic at the level of connections, is topology-agnostic, and takes into account both regional and global popularity of content types. We are aware that `ITMgen` has both advantages and disadvantages compared to GM. `ITMgen` is extensible; it can be easily extended as the dominant application mix of interdomain traffic changes, and data about new application types becomes available. We show how to parameterize `ITMgen` using mostly data that is available publicly. On the other hand, it might be challenging to parameterize and it describes only relative traffic between ASes. We are releasing `ITMgen` as a tool to enable researchers to generate synthetic, but representative traffic matrices for modeling and simulation purposes.

Part II: Microscopic view – price discrimination on the Internet

Chapter 4: Detecting Price and Search Discrimination on the Internet

In Part I we focused on measurements and modelling of interdomain traffic matrix, which is directly related with macroeconomics of the Internet. In this part we focus on describing phenomenas specific to Internet microeconomics.

The predominant economic model behind most Internet services is to offer the service for free, attract users, collect information about and monitor these users, and monetize this information. The collection of personal information is done using increasingly sophisticated mechanisms [48] and this has attracted the attention of privacy advocates, regulators, and the mainstream media. A natural question to ask is: what is done with all the collected information? And the popular answer is, the information is being used increasingly to drive targeted advertising.

Another hypothesis put forward for the wide-scale collection of information, and the related "erosion of privacy" is to facilitate price discrimination [64]. Price discrimination[1] is defined as the ability to price a product on a per customer basis, mostly using personal attributes of the customer. The collected information can be used to estimate the price a customer is willing to pay. Thus, it can have a huge impact on the e-commerce business, whose estimated market size is $961B [42]. The question we deal with in this chapter is, "*does price discrimination, facilitated by personal information, exist on the Internet?*". In addition to price discrimination, users can also be subjected to search discrimination, when users with a particular profile are steered towards appropriately priced products.

Detecting price or search discrimination online is not trivial. First, we need to

[1]We use the terms "price discrimination" and "search discrimination" because these terms are used in the economics literature to describe these phenomena; we are not taking a position on whether these phenomena are harmful or unethical.

decide which information vectors are relevant and can cause or trigger discrimination, if it exists. We look into three distinct vectors: technological differences, geographical location, and personal information (Sec. 4.2). For system based differences, the question is whether the underlying system used to query for prices make a difference? For location, we check whether the price for exactly the same product, sold by the same online site at the same time, differs based on the location of the *originating* query. And for personal information, we are interested if there is a difference in prices shown to users who have certain traits (affluent vs budget conscious). Second, we need to be able to finely *control* the information that is exposed while searching for price or search discrimination, to claim *causality*. In order to uncover price/search discrimination while addressing these concerns, we develop a comprehensive methodology and build a distributed measurement system based on the methodology.

Using our distributed infrastructure, we collect data from multiple vantage points over a period of 20 days (early July 2012), on a set of 200 online vendors. Our main results are:

• We find *no* evidence of price/search discrimination for system based differences, *i.e.*, different OS/Browser combinations do not seem to impact on prices.

• We find price differences based on the geographical location of the customer, primarily on digital products, up to 166%—e-books and video games. In addition, we also see price differences for products on a popular office supplies vendor site, when the queries originate from different locations within the same state (MA, USA). However, we cannot claim with certainty that these differences are due to price discrimination, since digital rights costs or competition could offer alternative interpretations.

• When we use trained personas that possess certain attributes (affluent, budget conscious), we find evidence of search discrimination. For some products, we observe prices of products that were shown to be up to 4 times higher for affluent than for budget conscious customer. We also observe this on a popular online hotels/tickets vendor.

• We find evidence of price discrimination when we consider the origin URL of the user. For some product categories, when a user visits a vendor site via a discount aggregator site, the prices can be 23% lower as compared to visiting the same vendor site directly.

4.1 Background

Price Discrimination. Price discrimination is the practice of pricing the same product differently to different buyers, depending on the maximum price (reservation price) that each respective buyer is willing to pay. For example, Alice and Bob want to buy the same type of computer monitor and visit the same e-commerce site at approximately the same time. Alice receives $179 as price while Bob gets $199. The seller offers different prices to them by profiling them (see Sec. 4.2.4 for details) and realizing that Alice has already visited many electronics' web-sites and therefore might be more price sensitive than Bob.

From an economics point of view, price discrimination is the optimal method of pricing and increases *social welfare* [81, 19, 54]. Despite its theoretical merits, buyers generally dislike paying different prices than their peers for the same product/service. From a legal point of view, the Robinson-Patman Act prohibits price discrimination in the US under certain circumstances [10] but the possibility is largely open in the current largely unregulated cross-boarder electronic retail market on the Internet. Recently, a new congress bill aims to make price discrimination on the Internet transparent to end users [76].

Historically, price discrimination has been practiced in myriad industries such as the US railways in the 19th century, flight tickets, personal computers and printers, and colleges fees [64]. Besides these examples, some minor instances of price discrimination have emerged in the last decade on the Internet as well, *e.g.*, Amazon showed different prices to customers [77], and more recently, Orbitz displayed search results in different orders to some group of customers [78]. We emphasize that price discrimination and price dispersion[2] are different concepts. Price dispersion occurs when the same product has different prices across different stores for reasons other than the intrinsic value of the product, *e.g.*, because one store wants to reduce its stock or has had a better deal with the manufacturer.

Search Discrimination. Another way to extract more revenue from buyers

[2] http://en.wikipedia.org/wiki/Price_dispersion

with a higher willingness to pay is to return more expensive products when they search within a product category. Search discrimination is different from price discrimination because instead of operating on one product, it operates on multiple products trying to *steer* buyers towards an appropriate price range. Ranking of search results greatly impacts the result eventually chosen by the user; users seldom go beyond the first page of results [45]. Hence the search provider, whether a generic search engine or search on e-commerce sites, is in a position enable such discrimination. For example, Alice and Bob are searching for a hotel in Redmond during the same days and for the same type of room. Their searches are launched at approximately the same time. A booking site offers Alice three hotels with prices $180, $200, and $220, while Bob receives quotes from a slightly different set of hotels with prices $160, $180, and $200. This can happen if the site has access to historic data that indicates that Alice tends to stay in more expensive hotels, or by other means such as system information [78]. While search personalization is not entirely new[3], in this chapter we draw attention to the *economic* ramifications of it, and in particular study if the information vectors that cause price discrimination also play a role in search discrimination.

Information leading to discrimination. In order to detect discrimination—price or search—we first need to fix the different axes along which the discrimination can take place. We consider three distinct sources of information:

• *Technological/System based differences:* Does the combination of OS and/or browser lead to being offered different prices?

• *Geographic Location:* Does the location of the originating query for the same product and from the same vendor/site play a role? Note that we are *not* interested in the same product sold via local affiliates—for instance Amazon has sites in multiple countries, often selling the same products.

• *Personal Information:* Does personal information, collected and inferred via behavioral tracking methods, impact on prices? For instance, does an 'affluent' user see higher prices for the same product than a 'budget-conscious' user?

Requirements of the system. Based on the definition of price and search discrimination, as well as the axes along which we seek to uncover discrimination, we set the following requirements for our methodology:

[3]With new implications being discovered, for instance the Filter Bubble concept [37]

• *Sanitary and controlled system:* In order to attribute *causality*, we need to have clean, sanitary, and controlled systems. We should be able to test for one of the axis described above, while keeping the others fixed. For all our measurements, we keep time fixed, *i.e.*, request all price quotations at nearly the same time.
• *Distributed system:* In order to have indicative results, we need a distributed system where we can collect measurements from multiple vantage points.
• *Automated:* To scale the study in terms of customers and vendors we need to automate the process.

4.2 Methodology

The test that we employ while searching for price discrimination is to select a website, an associated product, and then study whether the website returns dynamic prices based on who the potential buyer is. In all the experiments, we compare the results (price or search) retrieved simultaneously to exclude the impact of time from the analyses, *i.e.*, all measurements for a single product happen within a small time window.

4.2.1 Generic measurement framework

We have developed a measurement framework that uses three components: browsers, a measurement server, and a proxy server.[4] The browser(s) run on separate clean local machines, with the possibility to run over different OSes. To access the pages, we use a Javascript (JS) application that loads the pages in separate iFrames. We use browsers and JS to ensure we can browse sites that need full features (as opposed to issuing `wget`'s) and to ensure cross-browser compliance. The measurement server controls the JS robot.

Role of the Proxy. We used a proxy for three reasons: (i) We are interested in extracting prices embedded in the pages. Unfortunately JS cannot access and store the content of the opened pages due to its internal *Same Origin Policy*. Hence we configured the browsers to use the proxy server. The proxy then monitored and stored all the traffic going through it. (ii) Some of the destination sites

[4]We modified Privoxy [7].

(*e.g.* `amazon.com`) did not open in an iFrame by setting `X-Frame-Options` in the HTTP response headers. The proxy modified the headers on the fly so the option was removed before the page reached the browser. (iii) The proxies allowed us to add additional privacy features, *e.g.*, set the *Do Not Track* option in HTTP headers. In order to mimic behavior of users for sites that need interaction, we used iMacro [44].

Ensuring a Sanitary Environment. We made an effort to prevent any permanent data from being stored in the browser, and thus allowing tracking of the user. The proxy layer allowed us to remove the "Referer" field in the HTTP header that would point to the measurement server, and block pixel bugs [7]. All the browsers were configured to block 3rd party cookies, commonly used for tracking, and we also dealt with flash cookies, *etc.*. Additionally, after each measurement round we deleted the files that might have stored the browsers' state. This restrictive configuration was used for both the system- and the location-based studies.

4.2.2 System-based measurement specifics

We compared prices of various products accessed from different browsers running on different OSes, from a single geographical location (Barcelona, Spain). We used three systems: Windows 7 Professional, Ubuntu Linux 12.04 and Mac OS X 10.7 Lion with browsers: Firefox 14.0, Google Chrome 20.0 (for all the systems), Safari 5.1 (for OS X) and Internet Explorer 9.0 (Windows). Since we have fixed time and location and prevented identity information leakage, we attribute price difference to the employed system.

4.2.3 Location measurement specifics

To investigate the impact of a customer's geographical location on the prices she receives, we deployed several proxy servers at different Planetlab nodes. We chose 6 distinct sites: two sites in US (east and west coast), Germany, Spain, Korea, and Brazil. For this experiment, we used 6 separate, identical virtual machines with Windows 7 and Firefox. With this configuration, the only information that distinguished the browsers externally was their IP. We assume that the IP address is enough to identify the geographical location of the originating query and is

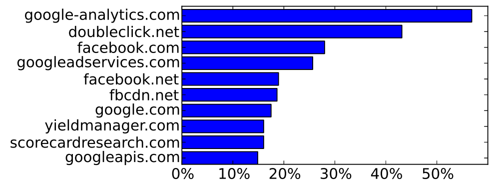


Figure 4.1: Presence of third party resources on the sites used for training personas.

enough for price discrimination to take place. We fixed time when we conducted our measurements across sites, syncing various sites using NTP.

4.2.4 Personal info measurement specifics

In order to uncover discrimination based on personal information, we follow two methods that differ in the amount of information that they employ. In the first we train "personas" that conform to two extreme customer segments: *affluent customers* and *budget conscious customer*. The two profiles are quite distinct. The budget conscious customer visits price aggregation and discount sites (like `nextag.com`). The affluent customer visits sites selling high-end luxury products. The customers might be tracked by third party aggregators (*e.g.*, DoubleClick) that have presence on many sites around the web and can chain such visits to construct a profile of the user.

We train personas as follows. We obtain the generic traits followed by an affluent consumer and a budget conscious consumer from [21]. An affluent consumer is more likely to visit "Retail–Jewelry/Luxury Goods/Accessories" sites as well as "Automotive resources" and "Community Personals" sites than the average user. For each of these categories, we use Alexa.com and Google to select top 100 popular sites, and configure a freshly installed system to visit these sites, and to train the profile. In order to mimic a real human, we train only between 9AM-12PM and use an exponential distribution (mean: 2 min) between requests. We do the same to train the "budget conscious" consumer by using the relevant sites. We train both profiles for 7 days, and we permit tracking and disable all blocking. Note that we can train multiple personas resembling different segments—this is left for future work. We show the distribution of third party trackers on the sites we used for the training in Fig. 4.1.

The second method that we use to test for discrimination based on personal information uses the "Referer" header that reveals where a request came from. Therefore, if you come from a discount site or a luxury site the e-commerce site where you land knows about it and can use it as indication of your willingness to pay. We fix one location—Los Angeles, USA—and fix one system—Windows 7 with Firefox—to run the personal information related measurements.

Assumptions: For the three sources of price discrimination we are studying, we assume that the information vectors we use are sufficient in isolation for price discrimination to kick-in. In reality, a composition of different vectors may be needed for price discrimination. For instance, personas and a specific type of system configuration may be needed together for price discrimination. Composing different vectors and then testing for discrimination is left for future work.

4.2.5 Analyzed Products

To determine the types of products to focus on, we selected the product categories from Alexa. In total, we examined 35 product categories (*e.g.*, "clothing") and we choose 200 distinct vendors (*e.g.*, `gap.com`). From the identified e-commerce sites, we selected 3 concrete products with their unique URLs (*e.g.*, specific piece of clothing). For each vendor, we selected low/mid/high price products. In case of hotels, we selected three different dates (low/mid/high season) at multiple locations. The 200 odd vendors we chose may appear to be a small set. However, we limit ourselves to 200 to first understand issues with scaling. In addition, these 200 vendors also account for the vast majority of user traffic as they include large vendors like `amazon.com` and `bestbuy.com`. We intend to increase these 200 vendors to 1000+ vendors to also cover long-tail sites. In the end we had a total of 600 products.

4.3 Empirical results

4.3.1 System based differences

We collected extensive measurements on 600 different products. We used the 8 distinct system–browser setups to examine the potential price differences. We ran the measurements for four days, and collected over 20,000 distinct measurement points in total. In addition, we queried Google and Bing to examine if the search results differ based on the systems. For this, we used 26 different phrases related to the products we analyze. The measurement did *not* reveal any price differences between the end systems. Regarding search discrimination, although we noticed slight deviations in the ordering of search results that were neither significant nor reproducible.

4.3.2 Geographic location

Next, we looked into the impact of geographic location from where the user accesses an e-commerce site. We issued queries through the proxies described in Sec. 4.2.3 on the same set of products/sites as before. In total, we accessed each product 10 times. The measurement results do *not* indicate significant differences, neither in prices nor in search results, for the majority of the products. However, the prices shown by three particular websites appeared to depend strongly on the users' location. In particular, `amazon.com` and `steampowered.com` returned prices for digital products (e-books and computer games, respectively) and `staples.com` for office products that differ between buyers at different locations.

In the case of Amazon, we observed price differences only for Kindle e-books. We queried the prices of books listed on the top 100 list of Amazon from six locations.[5] Only 27 out of these 100 books were available for purchase in their original english version from Amazon.com (US site) to customers coming from all the 6 locations we were testing. We illustrate the price differences of these products in Fig. 4.2, where we plot the ratio of the products' prices using the prices in New York, USA as reference. In majority of the cases, the price difference is at least

[5] For both website, results for US/LA and US/NY overlap and are not shown.

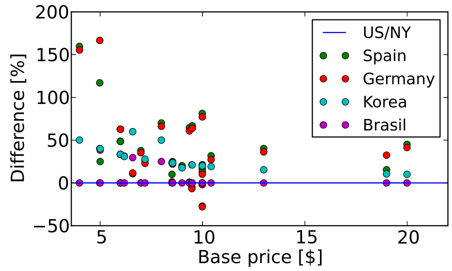


Figure 4.2: Price differences at Amazon based on the customer's geographic location using the prices in New York, USA as reference. For each of the considered products there exist at least two locations with different prices.

21%; however, in extreme cases it can be as high as 166%.

For the Steam site, we examined more than 300 additional products. We compared the prices of the products where their prices were displayed in the same currency to avoid the bias of currency exchange. We observed price differences for 20% of the products in case of Spain and Germany (figure not shown). Moreover, 3.5% of the products had different prices in case of US, Brazil, and Korea.

Next we analyzed the impact of location on a finer scale, *i.e.*, within the US only. We used 67 Planetlab nodes in US acting as proxy servers. We accessed 10 random products from `staples.com` using the proxies. 4 products showed different prices when accessed from different locations. In those cases, there were two distinct prices for the same product. We did not observe a significant correlation between the prices and population per state/city, population density per state, income per state, or tax rates per state.

We extended the study of `staples.com` by taking measurements within the same state (MA) to exclude inter-state tax differences. We selected 29 random products and 200 random ZIP codes.[6] Again, for 15 products the price varied up to 11% above the base price between the locations.[7]

Fig. 4.3 shows the price differences geographically. The values on the map show a mean price surplus calculated for a particular location over all the products. The map shows that the outskirts are shown higher prices than the large cities.

[6]When accessing `staples.com` from outside of US, the service asks for the customer's ZIP code, giving equivalent results as coming from a certain location.

[7]Base price - smallest observed price for a product.

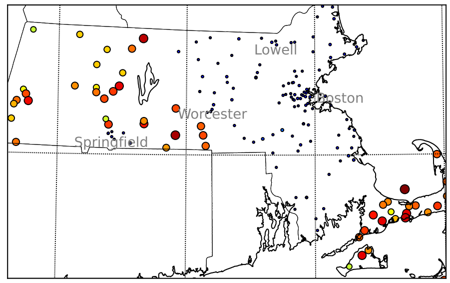


Figure 4.3: Price differences at **staples.com**. The dot sizes mark the mean price surplus for the locations, from 0% (small dots) up to 3.9% (large dots)

Discussion: Our system ensures that the only bit of information that is exposed is the IP address, hence the location. We see differences in prices for some digital goods as well as office supplies. We cannot claim to have discovered price discrimination since the differences might be attributed to other reasons such as intellectual property issues or increased competition between retailers or logistics. Further investigation is required on this issue.

4.3.3 Personal information

Trained personas. We used the previously trained personas (Sec. 4.2.4) to examine the discrepancies of products based on the browsing behavior. We also used a clean profile as a baseline. We did *not* observe price discrimination in our results; however, we observed different search results on two sites. First, we examined 12 search queries in `google.com`, three times for each profile. For half of the queries, the results included several suggested products, together with the prices. There is a noticeable difference in the prices of these products as we show in Fig. 4.4. For instance, the mean price was 4 times higher in case of "headphones" for the affluent persona than for the budget one. Second, we examined the top-10 hotel offers on Cheaptickets. We searched for hotels in 8 different cities on 8 different dates. The search engine of Cheaptickets returned offers with higher prices for the *affluent* profile (Fig. 4.5).

Originating web page. Our hypothesis for studying the origin is that the site that a customer uses to reach a product site can provide valuable information for pricing purposes. For example, if the customer comes from a discount site, she

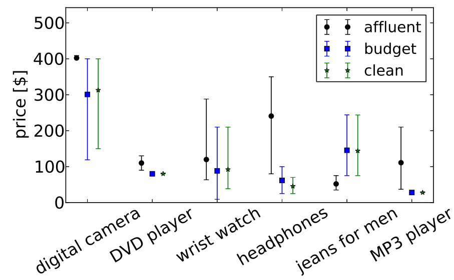


Figure 4.4: Prices (mean/min/max) shown by Google to the different personas. The median number of products in each category per persona is 12.

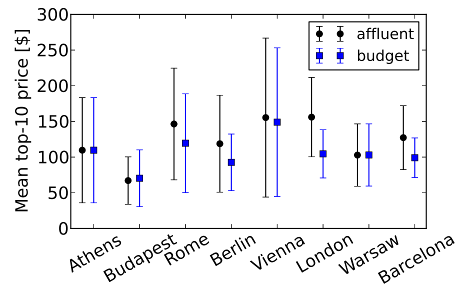


Figure 4.5: Mean prices (with std. deviations) of top-10 results from Cheaptickets.com returned to affluent and budget personas. The mean difference is 15%, and can be even as high as 50%.

will be more likely to be price sensitive than someone coming from a luxury site or a portal. Hence, we focus on price aggregator sites that provide a platform for vendors of various products and also provide discounts to users. We looked into a couple of aggregator sites (`nextag.com`, `pricerunner.co.uk`, `getprice.com.au`), but we only present results of one large site: `nextag.com`. We used a clean profile, with blocking enabled but enabled first party cookies. We examined 25 different categories of products available on `nextag.com`. We found two online vendors (`shoplet.com`, `discountofficeitems.com`) who returned different prices based on the originating web page of the customers. Both retailers specialize in office equipment. In case of `shoplet.com`, users get higher prices if they access a product directly via the retailer's website than when the price aggregator (`nextag.com`) redirects the user to the store. In the latter case, the aggregator redirects the user to an intermediate site that sets a cookie, and from this point on the user starts getting lower prices. We quantify the price differences with- and without

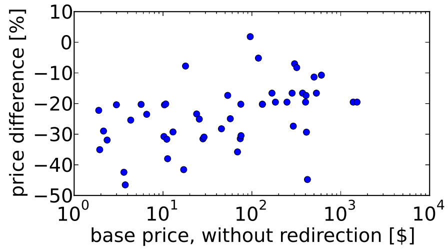


Figure 4.6: Price difference at the Shoplet.com online retailer site, with- and without redirection from a price aggregator.

the redirection in Fig. 4.6. The mean difference between the prices is 23%.

Discussion: We noticed signs of search based discrimination in case of trained personas. We stress that while we have not yet found price discrimination for trained personas, we did observe signs of discrimination via origin URL. We note that the entities who collect large amounts of information across the web (aggregators like Doubleclick)—and hence can create a more accurate representation of the user—do not actively engage in e-commerce. On the flipside, large vendors do not track users across the web. Thus, the entities who could utilize information of users for pricing are decoupled from those who collect such information. The redirection mechanism, that uses one bit of information, can be used effectively to narrow this information gap.

4.4 Related Work

Price discrimination is as old as retail itself [55] but online price discrimination is a fairly new phenomenon. To the best of our knowledge one of the first to conjecture the rise of online price discrimination driven by large scale collection of personal information was A. Odlyzko [64].

There is little prior work on price discrimination on the Internet. Beyond price discrimination, personalization of the web using personal information of users is an active area of research with the study of the filter bubble effect [39]. In contrast, we are interested in the economic implications of personalization – price discrimination on e-commerce domains. The notion of building large distributed systems to understand the effect of personal information on services obtained has been done for various reasons [38, 37]. Guha, *et al.* [38] focused on the impact of user characteristics on display advertisements. Our measurement framework presented in this work is similar – however, we focus on the differences of product prices instead of displayed ads. Our work is closely tied to online privacy, both in terms of usage of privacy preserving tools in our methodology, as well as implications of (loss of) privacy over price discrimination. For the former, we use the findings of Krishnamurthy, *et al.* [48] to block known forms of tracking, on our proxy as well as the browser. Besides cookies, other techniques can also uniquely identify users with high probability such as the properties of the browsers [31] and the browsing history [67], hence we take steps to counter such identification.

4.5 Conclusions

In this chapter we examined existence of Price Discrimination on the Internet. We looked at different information vectors that could possibly be used to trigger price discrimination, like technological differences or geographical location. We also examined impact of personal information using trained *personas*. In order to examine those information vectors, we followed rigorous methodology to control what information about the hypothetical "user" is released to the Internet, in order to be able to claim causality. We also built a distributed measurement system to collect the data. This system allowed us to examine prices of over 600 products offered by more than 200 on-line retailers, collecting over 20,000 measurement points.

The collected data revealed many examples of price differentiation. We found examples of differentiating prices based on geographical location of the customer for retailers like `steampowered.com`, `amazon.com` or `staples.com`, even though it cannot be unambiguously attributed to Price Discrimination since other factors (like intellectual property issues or logistics) might matter as well. Later we observed different search results presented to personas with certain attributes. In particular, we observed that more expensive products were presented to *affluent* than to *budget conscious* personas. We found examples of differentiating prices based on URL of origin, where a person reaching a vendor (e.g. `shoplet.com`) from a price aggregation (e.g. `nextag.com`) receives lower prices than a customer that reaches the vendor directly. The underlying assumption is that a person that accesses the retailer from the aggregator is more price-sensitive, thus it has lower reservation price.

In the next chapter we leverage this methodology and results to broaden the scope of the measurements using *crowd sourcing*. In particular, we release a system that helps the Internet users check if they are being discriminated.

Chapter 5: Using Crowd Sourcing to Detect Price Discrimination

In Chapter 4 we focused on finding empirical evidence that price discrimination indeed exists on the Internet. In this chapter we concentrate on broadening scope of the measurements and showing that *crowd sourcing* is a feasible method to enable wide-scale measurements on PD.

With the rise of e-commerce in the last decade many expected prices to move strictly in one direction – downwards – as a result of more intense competition fueled by the customers' ability to compare online the prices of different retailers. It was not long before the first concerns appeared with the conjecture that online shopping could backfire for customers in the form of price discrimination driven by the personal information of users collected by various online entities [64]. Such a possibility would further erode online privacy. For example, users frequenting luxury product websites or geo-located to certain ZIP codes could be tagged as affluent or price insensitive and consequently be displayed inflated prices.

We tested this conjecture in Chapter 4 and were able to demonstrate a few examples in which the prices of online offerings seemed to vary (please refer to [61] for concrete examples). In order to broaden the scope of our measurements so that we can derive general conclusions regarding the frequency and magnitude of suspected price discrimination, we turn to crowdsourcing. Crowdsourcing enables end-users to (i) point us to products and e-retailers that might be engaging in price discrimination, and (ii) aid us in extracting the prices of products from web pages without requiring manual intervention (Sec. 5.1). Crowdsourcing, therefore helps us in scaling up the search process. This is achieved by a browser extension called $heriff [60], (Sec. 5.2.1).

In this chapter we present the results obtained from $heriff collected for a three month period (Sec. 5.2.2). These results pointed to price variations observed in well known, but also in relatively unpopular sites and categories as well, different from our observations in [61], consistently over time and across different locations, underscoring the effectiveness of the crowdsourcing approach. We then perform a systematic measurement study of products on this set of e-retailers by performing a large crawl (Sec. 5.3) and understand the conditions that can lead to price variations Our main results include the magnitude of price variations for most e-retailers is between 10%–30%, the cheapest products often face the highest variation (×3) with the most expensive ones having lower variation (×1.5) , and physical location plays a role in price variations for different categories of products.

Note that there is little prior work in the area presented in this chapter. Therefore, for sake of conciseness, related work presented in Section 4.4 also applies here.

5.1 Setting the context

In this section, we set the context for our study by first discussing the questions we tackle, the challenges in answering these questions, and how we address them.

5.1.1 Open questions

- Do we see persistent, reproducible price variations and which e-retailers engage in price variation?

- How frequent and large are the observed variations? Which products experience price variations (cheaper or more expensive ones) and what type of variation (additive/multiplicative) do we see?

- Can we attribute price variation to actual price discrimination? In general, it is impossible to assert without access to the code that generates the prices that any price variation we observe is in reality price discrimination. However, we can eliminate several alternative causes that might explain them as discussed later.

- Finally, when there are price variations, can we attribute them to specific personal information traits (location, browsing history, *etc.*)?

5.1.2 Challenges

Any system wanting to perform large scale search for price discrimination has to parse product pages, extract the location of the price from web pages, and fan out queries to the same product page from other vantage points in order to compare the results.

The challenges that need to be addressed are as follows: (i) Different retailers have different web templates for presenting their products. Extracting the price of a product from an unknown template is non-trivial: a simple search for dollar or euro sign would fail since typically product pages include additional recommended or advertised products along with their prices. Therefore, for each retailer one needs to understand its template format and then write a specialized script for extracting the price. The problem with this is that it cannot scale with the number of retailers. (ii) Minimize noise as well as other possible reasons for price variations. Sources of noise include the retailer conducting A/B testing, timing difference between original and additional requests for comparison, and pricing format differences (different currencies, *etc.*). There are also reasons like taxation, logistics, shipping costs, intellectual property issues that can cause price differences that are not due to discrimination. For proper attribution of price discrimination, we need to ensure the known reasons cannot explain the variations. (iii) In order to better explain price discrimination, we need to control for factors like physical location, system issues, and browsing history.

Addressing challenges

To address scaling issues, we resorted to crowdsourcing, using *\$heriff* a browser extension for Firefox and Chrome. Crowdsourcing enables us to outsource the search for price variations and cover a larger part of the web. We describe the tool briefly in Sec. 5.2.1. The results from the tool uncover e-retailers that engage more in varying prices and this lets us focus more on these e-retailers, expanding scope and depth.

We took several steps in order to deal with noise. First of all, we synchronized the measurements from different vantage points so that they occur almost at the same time. This reduces the chance that an observed variation is because of time spread, availability, *etc.*. Also we repeated the same set of measurements multiple times to guarantee that the results are repeatable. This decreases the possibility of A/B testing and small-scale temporal effects being the cause of price variations.

Our different vantage points access always the same retailer site, but can be displayed prices on different currencies (the local one) because retailers typically geo-locate their IP address. We convert the prices obtained by the different vantage points for the same product into US dollars using the daily lowest and highest exchange rates. We keep only products whose price variation is strictly greater than the maximum gap that can exist given the two extreme exchange rates in our dataset. This guarantees that the observed price differences are not due to currency translation issues.

For factors like taxation, shipping costs, and custom duties, we manually checked to ensure these reasons cannot explain the price differences. Most e-retailers do not include shipping and taxing before checkout thus the great majority of our measurements was not affected by such issues. Custom duties are in most cases paid post sale directly between the customer and the custom authority without the intervention of the retailer.

5.2 Crowd-sourcing

In this section, we first describe the tool that was used to enable crowdsourcing and then detail the data we have collected using the tool. We end this section with an analysis of the collected data, which points to the retailers where price variations are prevalent, as seen by users around the world.

5.2.1 $heriff

We used a browser extension for Firefox and Chrome called *$heriff*. The extension performs the following tasks: (i) Enables the user to highlight a price of a product on an e-retailer, (ii) once the price is highlighted, the extension enables the user to check for price variations via a small click button, (iii) when the button is clicked, the exact URI is sent to 14 vantage points around the world where the same URI is requested and the entire webpage is downloaded, (iv) given the user has highlighted the price on the page, we use that information to extract the price from the downloaded page at different locations, (v) we send these prices back to the user from various locations. The user, therefore can observe if there are any variations for the exact product she searched for. Hence, the users have an incentive to return to *$heriff* time to time to check prices again. (vi) We store the pages for analysis in a database. The extension can be found at: `http://pdexperiment.cba.upc.edu/`.

As can be observed, we cannot control for the physical locations when the original query comes from, nor can we control for the system and/or the browsing history of the user who originated the query.

5.2.2 Collected data and analysis

We use a *crowdsourced dataset* collected by *$heriff* that contains 1500 requests (between Jan-May 2013) to check the prices of different products. The requests were issued by 340 different users from 18 countries. In total, the users of *$heriff* checked products from 600 domains. Afterwards, we systematically crawled the sites of retailers where *$heriff* revealed price differences. Before the analyses, we removed the noise from the crowdsourced dataset. Causes behind the noise include diverse number and date formats across countries, product customization not encoded on the URI, *etc.*. The *crawled dataset* focuses on 21 retailers. We randomly picked up to 100 products per retailer and checked the prices of these products on a daily basis for a week. The crawled dataset has 188K extracted prices in aggregate.

Which retailers return dynamic prices?

Fig. 5.1 lists the retailers with the highest number of instances of price variations in the crowdsourced dataset. The list includes a diverse set of sites that include bookstores, cloth retailers/manufacturers, office supplies/electronics, car dealers, department stores, hotel and travel agencies, *etc.*. For each one of these retailers, and for each one of the products checked on these retailers, we computed the ratio between the maximum and minimum price observed across the different measurement points. In Fig. 5.2 we plot the basic statistics (median,

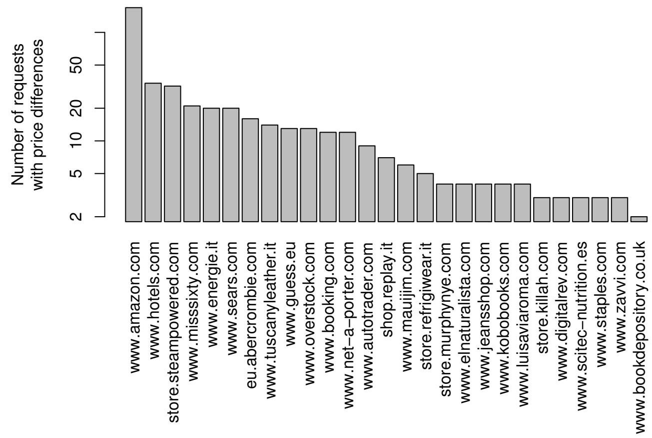


Figure 5.1: Domains with the highest number of request where price differences occurred

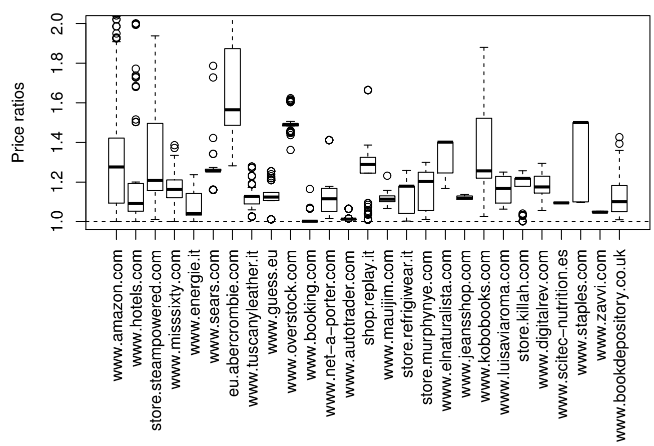


Figure 5.2: Magnitude of price differences per domains

25-, 75-percentile, and extreme values) of this ratio across all checked products in the dataset for each one of the retailers with the highest frequency of price variation. One can note that a variety of stores return prices that may vary anywhere between 15%-40% depending on the retailer, whereas there also exist few cases where the difference approaches a factor of $\times 2$! We note here that several of these retailers are not very popular (`www.elnaturalista.com`) and, in many cases, local (`store.refrigiwear.it`), underscoring the usefulness of crowdsourcing, as these retailers were not observed in previous studies [61].

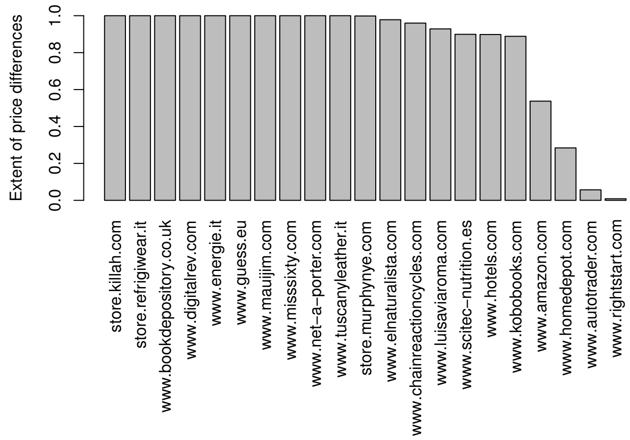


Figure 5.3: Measure extent of price variations for different domains

5.3 Crawling specific e-retailers

5.3.1 Retailers

Fig. 5.3 and Fig. 5.4 depict the same metrics with Fig. 5.1 and Fig. 5.2 but for the crawled instead of the crowdsourced dataset (Sec. 5.2.2). Fig. 5.3 shows the fraction of requests we sent out to each retailer that had price variation. In some cases, we see a 100% coverage, pointing to the fact that price variations are a persistent and repeatable phenomenon. Indeed, for the majority of retailers in the crawled dataset, we see the extent of price variation to be near complete (100%). In terms of the magnitude of price variability, Fig. 5.4 depicts values between 10% and 30% for most of the retailers—a non-trivial amount.

5.3.2 Looking into products

We now characterize price variations from the perspective of products. One open question is to understand if there is any correlation between the price of a product and the magnitude of the price variations associated with that product. For each product in crawled dataset (across all retailers) we compute the ratio between the maximum and minimum price across our measurement vantage points and plot them in Fig. 5.5 against the minimum observed price of each product. The figure

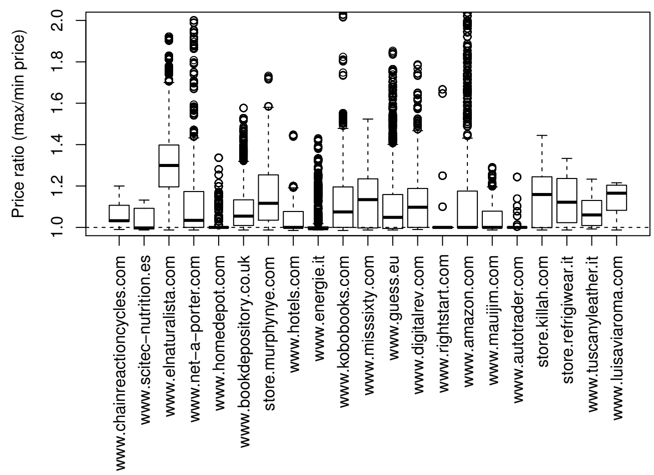


Figure 5.4: Magnitude of price variability per domain

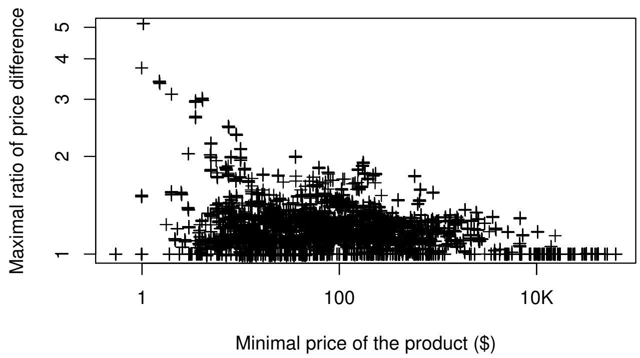
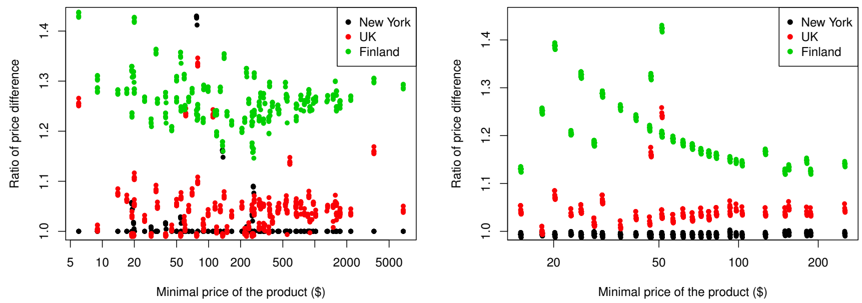


Figure 5.5: Maximal ratio of price differences per product price (all stores)

shows price differences occurring in the entire range of products costing from \$10 to \$10K. The highest differences are observed with cheaper products in the order of tenths of dollars, in which case differences up to $\times 3$ are observed. We also observe differences up to $\times 2$ for expensive products(in the \$1K range). For the most expensive products going into the multiple thousands, the price gap appears to be always smaller than $\times 1.5$.

In Fig. 5.5 the practices of a diverse set of retailers are mixed together. In order to unearth if there are difference strategies that are employed behind varying prices, we focus on individual retailers. In Fig. 5.6(a) we look at a retailer of photography equipment. For each one of the products from the retailer we studied, we plot a number of dots that is equal to the number of measurement points using



(a) `www.digitalrev.com` (b) `www.energie.it`

Figure 5.6: Ratio of price differences per product price

different colors to indicate each one of the vantage points. The x-axis denotes the minimum price of the product across all locations whereas the y-axis denotes the ratio between the price at the location of the dot and the minimum price. One can see parallel (to the x-axis) lines of different colors. This in effect means that the price variations between locations is *multiplicative*, equal to the gap between two lines on the y-axis, and this applies for the whole range of products (cheap as well as expensive ones). In Fig. 5.6(b) we show the same information from a clothes manufacturer. In this case we see a similar behavior for all but one location (green color). In that location the prices vary by an *additive* term compared to other locations. As the products become more expensive, the effect of the additive terms is progressively eliminated and the lines become parallel from $100 and onwards. We have other examples of retailers that apply a mix of multiplicative and additive pricing across our vantage points.

5.3.3 Does location have an impact?

Next we focus our attention on location. At a high level the question that we want to answer is whether users from certain locations tend to pay more for the same product than others. As with our previous analysis around products, we begin by showing aggregate results across all the retailers we focused on. For each product we compute the ratio of its price at a certain location over the minimum price across all locations for the same product. In Fig. 5.7 we present box-plots summarizing the main statistics of the above ratio for each one of the locations

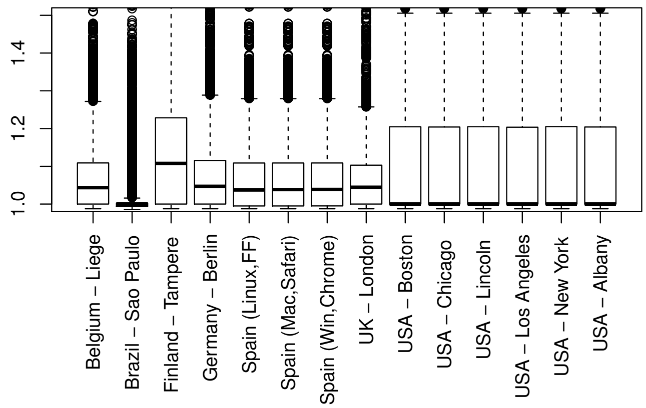
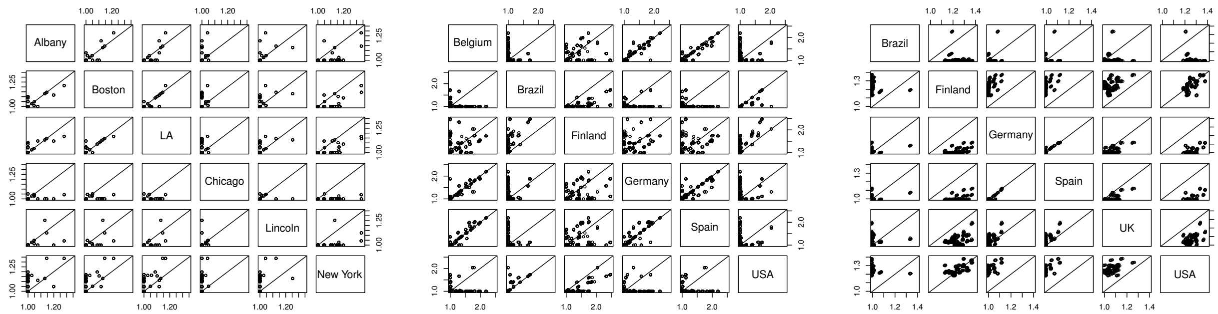


Figure 5.7: Magnitude of price differences per location (all)

where we had a measurement vantage point. From a first glance it seems that locations in USA and Brazil tend to get lower prices than locations in Europe. Within Europe, Finland stands out as the most expensive location.

To delve deeper into the effect of location we will refine the presented results by (i) focusing on specific retailers, and by (ii) presenting pair-wise comparisons of how a retailer prices its products at two different locations. We start with a retailer of home improvement appliances and equipments and look at its pricing across 6 US cities (Albany, Boston, LA, Chicago, Lincoln, New York). Fig. 5.8 (a) presents a grid of pairwise comparison subplots. The y-axis for each plot corresponds to the location represented in the row, while the x-axis for each plot represents the location shown in the column. For example subplot(1,2) has Albany on the y-axis and Boston on the x-axis. Within a subplot there exist points that correspond to individual products of the said retailer. The y-axis denotes the ratio between the price of the product at the y-axis location of the subplot and the minimum price of the product across all locations where we have vantage points. The x-axis denotes the same ratio with respect to the x-axis location of the subplot. Given these definitions, it is easy to note that a subplot where most of the dots fall along the main diagonal of the subplot signifies two locations that get similar prices from the said retailer across its products. If the dots cluster closer to the y-axis, then this is a sign that the y-axis location is more expensive than the x-axis location and inversely if the dots cluster along the x-axis.

With the above in mind we can identify a diverse set of pricing relationships.



(a) `www.homedepot.com` (b) `www.amazon.com` (c) `store.killah.com`

Figure 5.8: Magnitude of price difference per location

For example, we see that LA and Boston (subplot(3,2)) get similar prices, since most of the dots are aligned across the main diagonal (similarly with Albany and Boston (subplot(1,2) or (2,1)). On the other hand there exist examples where one location observes higher prices than the other – New York for example, appears to be consistently more expensive than Chicago (subplot(6,4)). There also exist mixed cases of pairs where one location is more expensive for some of the products but cheaper for some others, *e.g.*, Boston and Lincoln (subplot(2,5)). It is interesting to note that with different retailers these relationships change. Also, there exist retailers that have constant prices across US but vary them across countries, for example `amazon.com`, whose pairwise grid is shown in Fig. 5.8 (b). A diverse set of behaviors include equal price, more expensive/cheaper, and mixed can be observed across different countries. A third example from a clothes retailer is depicted in Fig. 5.8 (c).

In both the aggregate plot across all retailers (Fig. 5.7) as well as in the specific retailers of Fig. 5.8, Finland appears to be getting consistently the higher prices among other locations. For this reason, we are tempted to examine whether this is indeed true across each and every retailer in the crawled dataset. For this reason we plot in Fig. 5.9 the ratio between the price in Finland and the minimum price across all locations, for all the retailers of crawled. The results indicate that Finland is almost never the cheaper location (exceptions with `mauijim.com` and `tuscanyleather.it`).

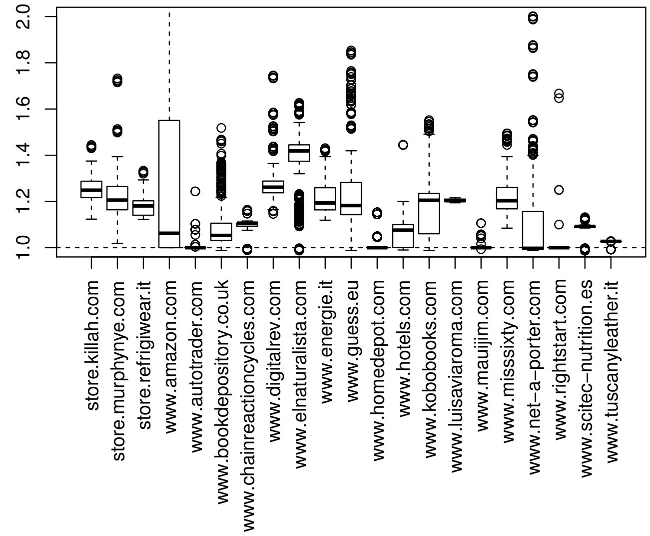


Figure 5.9: Magnitude of price differences per domains in Tampere, Finland

5.3.4 Personal information

In order to check if the personal information of users plays a role in price variations, we first train personas as described in an earlier paper [61]; we use an affluent and a budget conscious persona. We check for prices of different products at these specific e-retailers, taking measurements while keeping the location and time fixed, but we find *no* price differences.

We do, however find some price variations for Kindle ebooks on `www.amazon.com`, depending on if the user is logged in to the site or not. We present our results of collecting prices for three users with different profiles and compare that against the price observed when there is no login. Our measurements are conducted at the same time and from the same location, and are plotted in Fig. 5.10. We note price variations for the same product and it would appear there is little correlation to being logged in or not. There has been anecdotal evidence about `amazon.com` varying prices dynamically in the past [83], but for us to dig deeper for reasons is currently beyond the scope of this chapter.

As a first step towards understanding the mechanism behind varying prices and the parties that can possibly enable this, we investigate the frequency of third parties that are present on the retailers we study. It would appear that

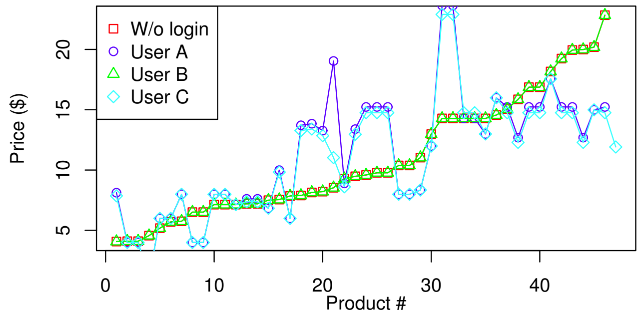


Figure 5.10: The impact of login on the price of Kindle ebooks at www.amazon.com

Google is present on most e-retailers with their analytics (95%) and doubleclick (65%) domains. Social networks have also significant presence on the retailers' sites through their widgets: Facebook (80%), Pinterest (45%), and Twitter (40%). While we do *not* see browsing history leading to price variations, it would be relatively easy for popular third parties to assist in price variations, fueled by the information they collect across the web. We leave this to future work.

5.4 Conclusions

In this chapter we demonstrated that crowd-sourcing is a feasible approach to help Internet users detect if they are subject to price discrimination. We also demonstrated that crowd-sourcing can be used to analyse long tail of the retailers. We presented results collected using dedicated browser extension – $heriff. Later we discussed results obtained using a large crawl on retailers indicated by $heriff. We also followed the rigorous methodology developed in Chapter 4 in order to reduce the measurement noise.

Using $heriff we collected data from 600 domains, indicated by the users from 18 countries. Next we focused on 21 retailers, collecting information for 100 products per retailer for a full week. Eventually this dataset consisted of 188K data points. The list of e-commerce websites that showed to differentiate prices includes bookstores, cloth retailers, office supplies, car dealers, department stores, hotel and travel agencies, etc. The differences were observed for products costing from 10 up to $10,000K. For instance, for the examined retailers the price variation between the minimum and maximum prices presented for the same product depending on user's location varies between 10% and 30%. We observed that the price difference can be either *additive* or *multiplicative*. We also identified and analysed a set of interesting pricing relationships between the particular retailers and selected locations in US and worldwide. We didn't find any significant differences of pricing for trained, *affluent* and *budget conscious* personas.

Chapter 6: Conclusions and Future Work

This chapter describes the conclusions drawn from the research presented in this thesis and proposes future directions where this work can be expanded.

The scarcity of the data necessary to study the interdomain traffic makes this research difficult, but not impossible. Although direct measures of the ITM are unlikely to be available, there is value in measuring qualitative properties of the ITM. In Chapter 2 we analysed properties of the ITM derived from GÉANT data. Practically every large scale dataset available to the researchers, even as large as the one from GÉANT, when compared with scale and variety of the Internet, cannot be called representative. Therefore, our research allows us to draw only limited conclusion about ITM, which is inherent to every research on wide-scale Internet based on a limited data sample. That said, cooperation with industry would allow to analyse other profiles of the traffic and would allow to explicitly draw boundaries of the research based on the traffic from research sources like GÉANT. In Part I we focused on spatial properties of the ITM which leave open questions about temporal properties. This research path could shed light on the long term evolution of the interdomain traffic matrix. Further works should also include large-scale research on the relations between the traffic matrix and the AS topology. Also in Chapter 2 we showed that there might exist a correlation between statistical properties of the traffic sourced by a network, and congestions inside that network. This suggests possibility to infer properties of the surrounding networks based on passive measurements of the traffic. Exploring this direction of the research would have a practical value for the network operators. Creating measurement tools that would give an insight into other networks could, for instance, affect peering decisions, optimize network operations, or give other business advantage.

As a next step we created a model that allows to generate synthetic snapshots of traffic matrices. A practical model should be a trade-off between the desired accuracy of the output and quality of the input. In Chapter 3 we show that openly available data combined with direct measurements can be used to generate synthetic snapshots of the traffic matrices. We acknowledge that static snapshots of the ITM might be insufficient in many applications – research on routing algorithms or network performance might require emphasis on temporal aspect of the traffic, which our model does not reproduce. On the other hand research on economics, pricing strategies, policy or peering strategies does not necessarily require the time series and we believe that is the area where our model can be applicable. Nevertheless, further research should put emphasis on recreating temporal properties of the ITM, once those are uncovered.

In our work we took a topology-agnostic approach towards generating the traffic matrices. We acknowledge that there might exist correlations between the interdomain traffic and the underlying topology, which should be examined in the further research. It should be also examined how to generate synthetic topologies matching the synthetic ITM, while preserving selected properties of the real network topology.

In Chapter 3 we show an alternative to the existing solutions to generate synthetic traffic matrices. Nevertheless, research community still lacks an established, standard method to model the interdomain traffic matrix. We believe that our work, together with existing findings, bring us a step closer to this elusive goal.

In Part II we show empirical evidence that price discrimination exists in the Internet. In Chapter 4 we analysed different information vectors that could lead to PD. We did not find evidence that shows that the user's operating system or browser might lead to differentiation in prices. On the other hand, we found examples of differentiation based on geographical location, even within the same US state. We also found examples of changes is prices based on personal information. For instance, a price can depend on the URL of origin – a particular originating web page could be a hint for a retailer that the customer is price-sensitive.

In Chapter 5 we argue that crowd-sourcing is a feasible method to conduct large-scale experiments on PD. We present an on-line tool that allows the Internet user to examine if any arbitrary price he observes varies in different geographical locations, with different operating systems and browser combinations. A natural next step would be to scale the experiment to cover a larger user and product base. Conducting such a large scale experiment would depend on delivering a non-trivial system, backed with significant design and engineering effort. Letting the experiment into public domain would also create challenges in analysing the gathered data. A successful and truly scalable platform should contain a crowd-based mechanism both to collect and to assess the collected data. Such a system should include incentives encouraging users to collect and analyse the data, which could be achieved for example by gamification [28]. This way the system would not be bound by the researcher's effort to process the garnered information.

Although price discrimination is probably as old as the first financial transaction, Internet gives the unique opportunity to explore vectors of price discrimination and how the personal information that enables PD circulates in the Internet. Research on PD naturally complements the broad and emerging research area related to personal information in the Internet, and its impact on search results [68] or advertising. To this end, future work could include uncovering the technological and economical mechanisms behind this kind of data marketing. As the issue of personal information in the Internet is gaining public attention, it would be also reasonable to work with policymakers to define practical bounds of the price discrimination research.

Appendices

Chapter A: Effective Processing of Backbone Traffic to Detect Portscans

Conducting work presented in the previous chapters required processing vast amount of online and offline data. While processing offline data requires merely computing time, capturing and processing online data from a working backbone link requires an extra effort. Improper configuration of the monitoring equipment or using resource consuming algorithms will result in lost packets or truncated flows and eventually will reduce usefulness of such data. In order to gain practical knowledge on how to handle such traffic, we performed an early exercise on detecting malicious activity in university backbone link. The results of our work on detecting portscans are presented in the following chapter. The tools and methodology developed during this work were used in the other parts of the thesis.

A.1 Introduction and Related Work

Every day both individuals and companies depend more on the reliability and safety of Internet connections. However, even today, entire industry branches or countries can be a target of an attack (e.g., `Stuxnet` [12]). Most attacks start with a recognition phase, where an attacker looks for attack vectors in one or several victim systems. Port scanning is arguably the most widely used technique by both worms and human attackers to probe for vulnerabilities in Internet systems.

Given the large implications in network security, several previous works have addressed the problem of how to efficiently and reliably detect port scans. Most proposed solutions require tracking individual network connections (e.g., [47, 75, 72]). This approach however does not scale to very high-speed links, where the number of concurrent flows can be extremely large. For example, a naive solution based on a hash table would require large amounts of DRAM (e.g., to store flow identifiers) and several memory accesses per packet (e.g., to handle collisions). Nevertheless, access times of current DRAM technology cannot keep up with worst-case packet interarrival times of very high-speed links (e.g., 32 ns in OC-192 or 8 ns in OC-768 links).

Traffic sampling is considered as the standard solution to this problem. Unfortunately, recent studies [51, 24] have shown that the impact of sampling on portscan detection algorithms is extremely large. Another alternative is the use of probabilistic, space-efficient data structures, such as Bloom filters [82, 62], which significantly reduce the memory requirements of detection algorithms. This way, the required data structures can fit in fast SRAM, which has access times below 10 ns. Although we are not aware of any survey paper covering the use of Bloom Filters for portscan detection, [62, 75] provide a good overview on the work in this area.

In this chapter, we present a practical method to detect TCP port scans in very high-speed links that follows this second approach. A key assumption behind our method is that, apart from data traffic, we can even discard most TCP handshake packets and still be able to successfully detect port scans.

First, we ignore legitimate handshakes using a *whitelist* of active server IP-port pairs. Second, we discard those failed connections that do not correspond to scans, such as TCP retransmissions, packets from other network attacks (e.g., SYN floods) or configuration errors (e.g., P2P nodes down or misconfigured domain servers). In order to discard handshake packets, we use two Bloom filters. Surprisingly, we show that this simple solution can drop about 85% of all handshake packets with negligible loss in accuracy. This significantly reduces the number of memory accesses, CPU and memory requirements of our algorithm.

After filtering most part of the traffic, we still need to track the number of failed connections for the remaining sources. Although there is a potentially very large number of active sources, most of them will fail very few handshakes, while scanners will fail many. Thus, the detection problem can be seen as the well-known problem of finding the top-k elements from a data stream [53]. In order to efficiently detect port scans, we use an efficient top-k data structure based on the *Stream-Summary* proposed in [57], which has a constant memory usage.

We evaluated our algorithm in 1 and 10 GigE academic networks [4]. Our results show that our method requires less than 1 MB to accurately monitor a 10 Gb/s link. Therefore, it can be implemented in fast SRAM and integrated in router line cards, or reside in cache memory of general-purpose processors.

The rest of this chapter is organized as follows. Sec. A.2 describes our portscan detection algorithm in detail. Sec. A.3 evaluates the performance of the algorithm with both packet traces and live network traffic. Finally, Sec. A.4 concludes the chapter and outlines our future work.

A.2 Detection Algorithm

Port scans are characterized by a simple feature: they attempt to connect to many targets but only get few responses. This imbalance in the number of attempts and successes is the basis of several portscan detection algorithms. A portscan detection algorithm can then be divided into two different problems: (1) detecting failed connections, and (2) tracking the sources responsible for them. Both (1) and (2) are challenging in high-speed networks, since they require a significant amount of memory and computing power to process packets at line speed. As already discussed in Sec. A.1, a naive solution based on a hash table is impractical in this case, although it can be used in small networks.

In this section, we present a practical solution that copes with these two problems by reducing both the volume of processed traffic and the memory requirements of the detection algorithm. In Sec. A.2.1, we describe a simple method to discard unnecessary traffic using Bloom filters, which significantly simplifies problem (1), while Sec. A.2.2 concentrates on identifying scanners using a lightweight counting structure that addresses problem (2).

For the sake of clarity, throughout this section, we will refer to the client host that initiates the handshake as A, with IP address A_{ip}, and to the server that receives the connection as B, with address B_{ip} and port B_{port}.

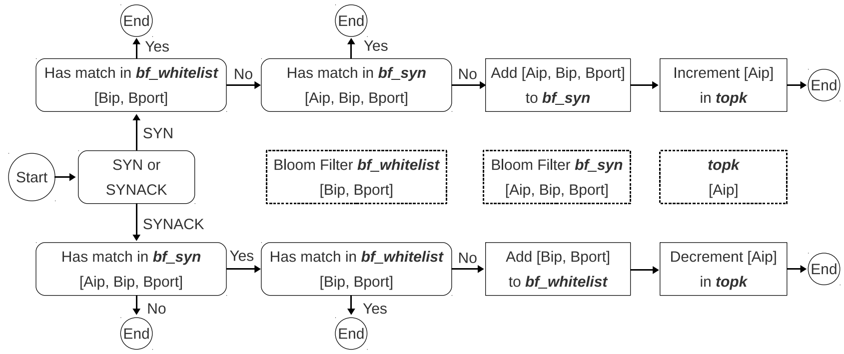


Figure A.1: Algorithm description.

A.2.1 Detecting Failed Connections

We can define a failed connection as one for which a client does not get a *SynAck* response from the server after having sent the corresponding *Syn* packet. Therefore, to detect failed connections, we can ignore data traffic and focus only on *Syn*/*SynAck* packets. According to our traces (described later in Sec. A.3), these control packets represent only 1.5% of all TCP traffic.

In addition, we can ignore legitimate handshakes to detect port scans, given that a scanner will always fail a large number of connections compared to a normal host. In order to efficiently discard connections directed towards a working service, we can use a Bloom filter that maintains a whitelist of active server IP-port pairs (*bf_whitelist*). In particular, for every new *SynAck* response, we add the tuple $[B_{ip}, B_{port}]$ into this Bloom filter.

Since we are especially interested in those clients that connect to many unique destination addresses and ports, we can also discard those repeated connection attempts to the same destination. Besides standard TCP retransmissions, many applications try to reconnect several times (even hundreds) to the same destination after a failed connection (e.g., P2P nodes, misconfigured proxies, mail servers or VPN applications). Surprisingly, repeated *Syn* packets are extremely common according to our traces (see Sec. A.3). In order to efficiently drop duplicated *Syn* packets to the same destination IP-port pair, we use a second Bloom filter (*bf_syn*). For every *Syn* packet observed, we store the tuple $[A_{ip}, B_{ip}, B_{port}]$ in the Bloom filter. As we will see later, using this second filter has the additional advantage of protecting the *bf_whitelist* from being saturated by many *SynAck* packets sent by a malicious user (i.e., *SynAck* packets are ignored if they are not an answer from a previous *Syn*).

Although Bloom filters can have false positives, they have a negligible impact on our method as we show in Sec. A.3. In addition, in case that one or both filters get saturated (e.g., if they are not properly dimensioned), the algorithm will produce False Negatives instead of False Positives, which is an important feature for systems automatically blocking port scanners [82].

Fig. A.1 presents our algorithm in detail. After a packet arrival, we check if it is a *Syn* or a *SynAck* packet. Otherwise, the packet is dropped. In case it is a *Syn* packet, we check if the $[B_{ip}, B_{port}]$ tuple corresponds to a known destination in the *bf_whitelist*. In this case, the packet is directly dropped. If not, we check if it is a repeated connection attempt in the *bf_syn* filter. In this case, the packet is also dropped. Otherwise, the $[A_{ip}, B_{ip}, B_{port}]$ tuple is stored in the *bf_syn* filter and the A_{ip} source is incremented in the counting structure (described later in Sec. A.2.2). For a *SynAck* packet, we first check if it is a response from a previous *Syn* packet in the *bf_syn* filter. Otherwise, the packet is dropped. Next, we check if the $[B_{ip}, B_{port}]$ tuple is already in the *bf_whitelist*. If not, the destination $[B_{ip}, B_{port}]$ is stored in the whitelist and the $[A_{ip}]$ source is decremented. Therefore, we use the *bf_whitelist* for two different purposes: (i) to keep track of active destinations, and (ii) to check if a source needs to be decremented after the connection has been established.[1]

A.2.2 Identifying Scanners

The algorithm described in Sec. A.2.1 produces a series of increments and decrements for new connections and completed handshakes respectively. From this sequence, we want to identify the most active producers of failed connections, which will very likely correspond to port scanners. This can be seen as the well-known problem of identifying the top-k most frequent elements in a data stream.

[1]Note that using *bf_whitelist* to check which decrements are needed can introduce errors of 1 unit in the counting structure if several *Syn* packets from different sources are sent to an active destination before it enters the whitelist. Although this unusual situation cannot be exploited by an attacker, it could be easily solved by adding a filter similar to *bf_syn* for *SynAck* packets.

For this purpose, we need a data structure that has limited memory usage and supports both incrementing and decrementing. Fortunately, the recent literature provides us with several efficient top-k algorithms [53]. From those, we selected the *Stream-Summary* data structure [57], since it uses a constant (and small) amount of memory. However, our algorithm is not bound to a particular top-k data structure. Although the original *Stream-Summary* does not support decrementing, we made a straightforward extension to support a limited number of decrements. We called this extension *Span-Dec*. As we will see in Sec. A.3, in the particular context of portscan detection, the data structure behaves almost like an ideal hash table, but using much less memory. Although the particular implementation details of the top-k data structure are not essential to understand our algorithm, for the sake of completeness, we include below a short description of both mentioned structures.

Stream-Summary. This structure is part of the *Space-Saving* algorithm [57] that finds the most frequent elements in a data stream. It is able to observe up to $elem_{max}$ distinct elements at once. Every element e_i has an assigned counter cnt_i. All counters with the same value are linked into the same bucket. The buckets are linked together and they can be dynamically created and destroyed. When an element e_i is incremented, it is detached from its bucket and attached to a neighbor bucket with the new value. When the maximum number of observed elements ($elem_{max}$) is reached, a new incoming element evicts the element with the smallest counter. Each element has a maximum overestimation ε_i that depends on the value of the evicted element. The element frequency is estimated as $freq(e_i) = cnt_i - \varepsilon_i$. The algorithm is lightweight and it requires only $\frac{1}{\epsilon}$ counters for a specified error rate ϵ. See [57] for a more detailed description.

Span-Dec. The original *Stream-Summary* does not support decrementing. However, we need to discount those established connections for which the corresponding *Syn* has passed both Bloom filters. Therefore, we made a simple modification to the original *Stream-Summary* to support a limited number of decrements. In particular, instead of having a single counter per element, we use two counters: $cnt_L(e_i)$ and $cnt_H(e_i)$. We also specify a maximum allowed difference between both counters $span_{max}$, which controls the tradeoff between the number of allowed decrements and the error ε_i of the estimate. When an element is incremented,

Table A.1: Statistics of the traces. `trace C` only accounts for *Syn/SynAck* packets.

	`trace A` 30min @ 1GigE	`trace B` 2h @ OC-3	`trace C` 30min @ 10GigE	`trace A0` 30min @ 1GigE
date	2010-05-18	2010-04-16	2010-07-29	2010-05-18
TCP packets	228,848,927	144,885,865	13,978,845	97,380,742
TCP sources	188,136	263,055	467,264	89,086
TCP flows	2,892,334	5,199,928	11,526,323	1,133,392
average usage	879.1 Mb/s	185 Mb/s	3.5 Gb/s	n/a

$cnt_H(e_i)$ is moved as in the original *Stream-Summary*. In case that the difference between both counters is greater than $span_{max}$, the $cnt_L(e_i)$ is also incremented. In order to decrement an element e_i, the $cnt_H(e_i)$ is decremented, but never below the value of $cnt_L(e_i)$. This solution can be understood as an "undo" operation, where $span_{max}$ is the "undo" depth. The frequency of an element e_i is estimated as $freq(e_i) = cnt_H(e_i) - \varepsilon_i$. The technical report [58] provides a detailed description of this extension.

As shown in Fig. A.1, our detection algorithm uses *Span-Dec* to maintain the count of failed connections per source $[A_{ip}]$. This solution is useful to detect both horizontal and vertical port scans. However, if we are interested only in a particular type of scan, we can use instead $[A_{ip}, B_{port}]$ to detect horizontal port scans and $[A_{ip}, B_{ip}]$ to detect vertical ones.

A.3 Results

In the evaluation we used four traces. `trace A` was captured from the 1GigE access link of UPC, which connects about 50,000 users. `trace A0` is a modified version of `trace A` that we describe later. `trace B` was taken from the MAWI Working Group Traffic Archive [5]. `trace C` was captured from the 10GigE link that connects the Catalan Research and Education Network to the Internet. This link connects more than seventy universities and research centers. Due to the link speed, for `trace C` we only collected *Syn/SynAck* packets. Statistics of the traces are presented in Tab. A.1. We published all the packet traces used in this work, with anonymized IP addresses, at [4].

For the evaluation, we needed a ground truth trace to check if a detected scanner was a real scanner or a (misclassified) legitimate source. For this purpose, we modified `trace A` by removing all real scanners. We scanned the trace using Bro [69] with both its standard algorithm and the TRW algorithm. Although Bro is an online tool that does not guarantee an accurate ground truth, we used a low alarm threshold (25) and removed all the flows from the reported IP addresses to make sure that no scanning traffic is left, even if some legitimate traffic was also removed. Later, following the methodology proposed in [62], we injected artificial scans to build a ground truth: 1000 scanners with success ratio 0.2 and 1000 benign sources with success ratio 0.8. The interval between *Syn-SynAck* packets was taken uniformly from the range $(0, 450ms)$, while the backoff time between *Syns* was modeled using an exponential distribution [62]. All modifications resulted in `trace A0` that serves as the ground truth for Sec. A.3.1. Traces `B` and `C` were not modified.

A.3.1 Evaluation

This section covers the evaluation of our algorithm. First, we present an example of how it is dimensioned. Next, we check the performance and validate its accuracy with packet traces. Finally, we deploy it in an operational 10 GigE link.

Dimensioning. We followed a conservative approach to handle an unexpected growth of traffic or peaks. For *bf_whitelist*, we checked the mean number of distinct $[B_{ip}, B_{port}]$ tuples in the trace, multiplied this value by 3 and we assumed a maximum collision probability of $p_{coll} = 0.01$. We used an arbitrary length of the measurement window of 2 minutes. Although in this chapter we do not evaluate this parameter, its value is important. As the filters are reset at the end of every period, the window size represents a tradeoff between the memory usage of the algorithm and its ability to detect slow scanners. With those values, we calculated the optimal size of the Bloom filter. We repeated the procedure for *bf_syn* using the unique number of $[A_{ip}, B_{ip}, B_{port}]$ tuples. The value of $span_{max}$ depends on the number of *Syn* packets concurrently sent by a source to distinct active destinations, which are not yet in the whitelist. We set this value according to 95th percentile of the traffic. For *topk* we arbitrarily set $elem_{max}$ to 10000 elements,

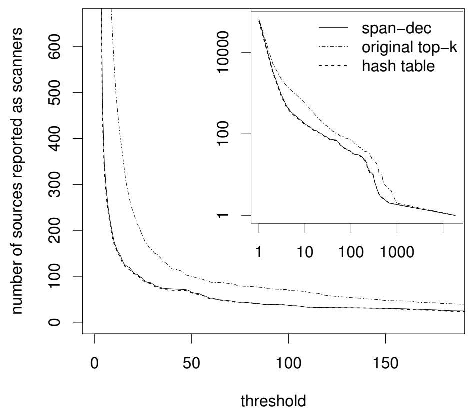
Table A.2: Configuration parameters for the evaluated traces.

	`trace A`	`trace B`	`trace C`	`trace A0`
bf_syn size	256KB	256KB	1MB	64KB
bf_whitelist size	128KB	128KB	512KB	32KB
$span_{max}$	6	4	10	5

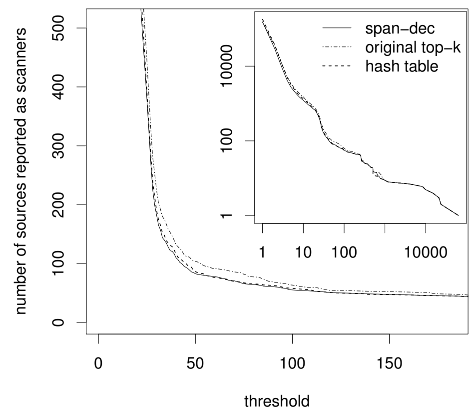
unless otherwise noted. Resulting parameters are presented in Tab. A.2. More details about the dimensioning procedure can be found in [58].

Detection threshold. To present the results for traces `A`, `B` and `C`, we follow the methodology used in [70]. Fig. A.2 depicts the results when running the algorithm on our traces with the parameters described in Tab. A.2. We plot the total number of sources reported as scanners as a function of the detection threshold. The threshold is the number of failed connections over which we classify a source as a scanner. The embedded plots show the whole range of data in a log-log scale, while the main plot presents only the part where the number of reported sources grows rapidly, in a linear scale. The "hash table" line presents the results obtained using hash tables to count distinct *Syn* and *SynAck* packets. In this scenario, all packets are counted with perfect accuracy. Results placed above this line indicate the presence of False Positives (FP), while those placed below the line imply False Negatives (FN). "Span-dec" line plots the results obtained when our counting structure was used. Both lines almost overlap indicating that our algorithm is close to an ideal tracking scheme using a hash table, but without its memory constraints. In particular, for high threshold values our algorithm features almost perfect performance. "Original top-k" shows the results obtained with the original *Stream-Summary* structure [57]. The large number of FP shows the necessity of supporting decrements in the counting structure.

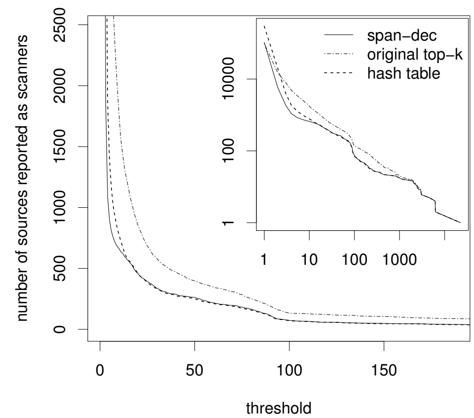
Accuracy. The results in Fig. A.2 were not enough to validate the actual accuracy of our algorithm. For this purpose, we used the ground truth `trace A0`, for which we knew the actual scanners and legitimate hosts. Our results show that, for thresholds higher than 20, the algorithm obtained perfect accuracy (i.e., 0 FP, 0 FN, and 100% detected scanners). More details about the accuracy of our algorithm and the impact of each configuration parameter are given in [58].



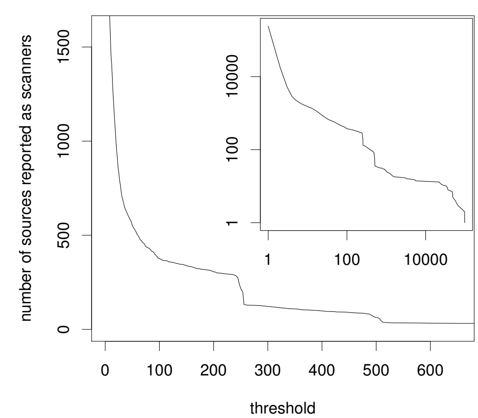
(a) `trace A` - 1 Gb/s UPC link



(b) `trace B` - MAWI traffic



(c) `trace C` - 10 Gb/s CESCA link



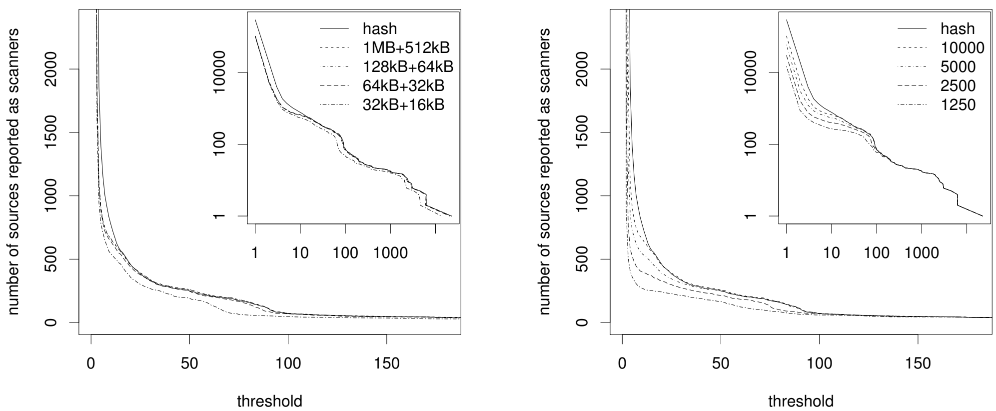
(d) online - 10GigE link

Figure A.2: Evaluation results on the traces - number of sources reported as scanners vs. detection threshold. Main graphs show a part of the data in a linear scale, embedded graphs show the whole range of data in a logarithmic scale.

Filter performance. Tab. A.3 presents the performance of the filters. The *Space usage* row shows the maximum space usage of each Bloom filter and (in brackets) the empirical collision probability. The probabilities are very small, even negligible. The *evictions* row shows the rate of traffic dropped by each filter (relative to the input packets of that filter). *Total packets evicted* gives the total ratio of handshake packets discarded by any of the two filters. Both filters together drop about 85% of all handshake packets. Thus, only 15% of all *Syn*/*SynAck* packets result in increments or decrements in the counting structure. Given that the counting error

Table A.3: Usage of the filters during the evaluation (evictions: *Syn* / *SynAck*)

	`trace A`	`trace B`	`trace C`
space usage: *bf_whitelist*	6.78% (6.59e-09)	1.90% (8.94e-13)	4.66% (4.77e-10)
space usage: *bf_syn*	13.27% (7.25e-07)	29.07% (1.75e-4)	11.02% (1.97e-07)
evictions: *bf_whitelist*	52.7% / 67.1%	24.7% / 76.2%	54.3% / 77.9%
evictions: *bf_syn*	61.2% / 65.0%	54.3% / 72.0%	55.4% / 64.2%
total packets evicted	84.3%	73.5%	84.4%



(a) filters (*bf_syn*+*bf_whitelist*) (b) max. number of elements in *topk*

Figure A.3: Impact of the memory size compared to an ideal scheme (`trace C`).

depends directly on the number of introduced elements, with a smaller number of entries we achieve better accuracy with less space.

Memory size. Finally, we evaluated the impact of the memory size on the accuracy of the detection algorithm using `trace C`. First, we examined the impact of the size of the Bloom filters using 10000 entries in the *topk* structure. Results are presented in Fig. A.3a. Filters below 96KB present FN due to collisions, as discussed in Sec. A.2.1. With filters of 192KB (128KB+64KB) and a threshold above 100, the algorithm performs very close to the optimal. Using these filters, we examined the influence of the maximum number of elements ($elem_{max}$) in the *topk*. The results are presented in Fig. A.3b. We can see that, for thresholds above 100, even with 2500 elements in the *topk* we still obtain very good accuracy. In our implementation, this configuration occupies only 417KB for a 10 GigE link.

Online deployment. In order to evaluate the real-time performance of the algorithm, we implemented it in the CoMo system [22] and deployed it on the 10GigE link from where `trace C` was collected. The hardware platform consisted of a PC with an Intel Xeon at 2.40GHz with two DAG 5.2SXA cards. A filter to discard non-*Syn*/*SynAck* packets was set in both cards. The filtering also can be done easily in software, since it requires only checking *Syn* and *Ack* flags in a TCP header. We run the program for 100 min. (13-12-2010 at 10:50). The average traffic in the link was 5.4 Gb/s. The CPU load was about 5% during the whole experiment. For both filters, the maximum usage was 18.5% with a maximum collision probability of 7.31e-06. The threshold-alarm graph is presented in Fig. A.2d.

A.4 Conclusions

In this chapter, we presented a practical approach to detect port scans in very high-speed links. The key idea behind our approach was to discard as much traffic as possible at early processing stages in order to reduce both the CPU and memory requirements of our algorithm. We used two simple Bloom filters that maintain a whitelist of active destinations and efficiently track TCP handshakes, and combined them with an efficient top-k data structure to track failed connections. Both Bloom filters together can early discard about 85% of all handshake packets in our traces.

Our evaluation with four traces from different scenarios showed that our algorithm can achieve almost perfect accuracy with very little memory. We also deployed our algorithm in an operational 10GigE link and showed that it can work online. Also, we made a new dataset available to the research community, so that our results can be validated and compared with other solutions.

Bibliography

[1] http://www.alexa.com.

[2] ITMgen tool and other resources (e.g., the marketing reports) can be found at http://monitoring.ccaba.upc.edu/itmgen.

[3] http://www.cesca.cat/en/communications/anella-cientifica. Regional research network and AS, containing universities and research units.

[4] UPC/CESCA traces: http://monitoring.ccaba.upc.edu/portscan/traces.

[5] MAWI Working Group Traffic Archive: http://mawi.wide.ad.jp/mawi/.

[6] ipoque. PACE: Network Analysis with DPI. http://www.ipoque.com.

[7] Privoxy, http://www.privoxy.org/.

[8] Sandvine Global Internet Phenomena Report: Fall 2011.

[9] Systems and methods for personalized pricing. http://patentscope.wipo.int/search/en/WO2013025536.

[10] The Robinson-Patman Act, Pub. L. No. 74-692, 49 Stat. 1526, 1936.

[11] CIA - The World Factbook. https://www.cia.gov/library/publications/the-world-factbook/r 2009. (accessed 2012).

[12] http://www.bbc.co.uk/news/world-middle-east-11414483, 2010.

[13] Boston Consulting Group report. https://www.bcgperspectives.com/content/articles/media_en 2012.

[14] How the journal tested prices and deals online. http://blogs.wsj.com/digits/2012/12/23/how-the-journal-tested-prices-and-deals-onlin December 2012.

[15] Websites Vary Prices, Deals Based on Users' Information. http://online.wsj.com/article/SB10001424127887323777204578189391813881534.html, December 2012.

[16] The messaging apps taking on facebook, phone giants. http://www.wsj.com/news/articles/SB10001424127887323466204578382733261211950, May 2013.

[17] Want a deal online? pose as a bargain shopper. http://blogs.wsj.com/digits/2013/01/10/want-a-deal-online-pose-as-a-bargain-sh January 2013.

[18] WhatsApp Shows How Phone Carriers Lost Out on $33 Billion. http://www.bloomberg.com/news/2014-02-21/whatsapp-shows-how-phone-carriers-los February 2014.

[19] A. Acquisti and H.R. Varian. Conditioning Prices on Purchase History. *Marketing Science*, 24(3), 2005.

[20] D. Alderson, H. Chang, M. Roughan, S. Uhlig, and W. Willinger. The many facets of internet topology and traffic. *Networks and Heterogeneous Media*, 2006.

[21] AudienceScience. http://www.audiencetargeting.com.

[22] P. Barlet-Ros et al. Load shedding in network monitoring applications. In *Proc. of USENIX ATC*, 2007.

[23] V. Bharti, P. Kankar, L. Setia, G. Gürsun, A. Lakhina, and M. Crovella. Inferring invisible traffic. In *CoNEXT*. ACM, 2010.

[24] Daniela Brauckhoff, Bernhard Tellenbach, Arno Wagner, Martin May, and Anukool Lakhina. Impact of packet sampling on anomaly detection metrics. In *Proc. of ACM SIGCOMM IMC*, 2006.

[25] M. Cha, H. Kwak, P. Rodriguez, Y.Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proc. of the 7th ACM SIGCOMM conference*, 2007.

[26] Hyunseok Chang, Sugih Jamin, Z. Morley Mao, and Walter Willinger. An Empirical Approach to Modeling Inter-AS Traffic Matrices. In *Proceedings of the Internet Measurement Conference (IMC)*, 2005.

[27] CNN. Web sites change prices based on customers' habits. `http://edition.cnn.com/2005/LAW/06/24/ramasastry.website.prices`, 2000.

[28] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara, and Dan Dixon. Gamification. using game-design elements in non-gaming contexts. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 2425–2428. ACM, 2011.

[29] Amogh Dhamdhere and Constantine Dovrolis. Ten years in the evolution of the internet ecosystem. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 183–196. ACM, 2008.

[30] A.B. Downey. Evidence for long-tailed distributions in the internet. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Moeasurement*. ACM, 2001.

[31] P Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies, LNCS 6205*, pages 1–18. 2010.

[32] V. Erramill, M. Crovella, and N. Taft. An independent-connection model for traffic matrices. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006.

[33] Wenjia Fang and Larry Peterson. Inter-AS Traffic Patterns and Their Implications. In *IEEE Global Telecommunications Conference (GLOBECOM)*, Dec 1999.

[34] A. Feldmann et al. A methodology for estimating interdomain web traffic demand. In *IMC*, 2004.

[35] A. Feldmann, N. Kammenhuber, O. Maennel, B. Maggs, R. De Prisco, and R. Sundaram. A Methodology for Estimating Interdomain Web Traffic Demand. In *Proceedings of ACM SIGCOMM Internet Measurement Conference (IMC)*. ACM, 2004.

[36] K. Gadkari, D. Massey, and C. Papadopoulos. Dynamics of prefix usage at an edge router. In *Passive and Active Measurement*. Springer, 2011.

[37] Georgia Tech Information Security Center. Filter Bubble project, 2012. http://bobble.gtisc.gatech.edu/.

[38] S. Guha, B. Cheng, and P. Francis. Challenges in measuring online advertising systems. ACM IMC '10.

[39] Aniko Hannak, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. Measuring personalization of web search. In *Proceedings of the 22nd international conference on World Wide Web*, pages 527–538. International World Wide Web Conferences Steering Committee, 2013.

[40] Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. Measuring price discrimination and steering on e-commerce web sites. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 305–318. ACM, 2014.

[41] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930 (Best Current Practice), March 1996.

[42] IMRG. B2C Global e-Commerce Overview 2012.

[43] International Telecommunication Union. The World in 2013: ICT Facts and Figures. http://www.itu.int/ITU-D/ict/facts/index.html, 2013.

[44] iOpus. iMacro. https://addons.mozilla.org/en-US/firefox/addon/imacros-for-firefo

[45] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. SIGIR '05.

[46] Sydney Jones and Susannah Fox. *Generations online in 2009.* Pew Internet & American Life Project Washington, DC, 2009.

[47] J. Jung et al. Fast portscan detection using sequential hypothesis testing. *In 2004 IEEE Symposium on Security and Privacy.*

[48] B. Krishnamurthy, D. Malandrino, and C.E. Wills. Measuring privacy loss and the impact of privacy protection in web browsing. ACM SOUPS '07.

[49] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. Internet inter-domain traffic. In *Proceedings of the ACM SIGCOMM 2010 conference*, 2010.

[50] Mathias Lécuyer, Guillaume Ducoffe, Francis Lan, Andrei Papancea, Theofilos Petsios, Riley Spahn, Augustin Chaintreau, and Roxana Geambasu. Xray: Enhancing the web's transparency with differential correlation. *arXiv preprint arXiv:1407.2323*, 2014.

[51] Jianning Mai, Ashwin Sridharan, Chen-Nee Chuah, Hui Zang, and Tao Ye. Impact of packet sampling on portscan detection. *IEEE J. Select. Areas Commun.*, 2006.

[52] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband Internet traffic. In *IMC*, 2009.

[53] N. Manerikar and T. Palpanas. Frequent items in streaming data: An experimental evaluation of the state-of-the-art. *Data & Knowledge Engineering*, 2009.

[54] R.P. McAfee. Price Discrimination. In *Issues in Competition Law and Policy, vol. 1.* 2008.

[55] Richard D McKenzie. *Why Popcorn Costs So Much at the Movies*. Springer, 2008.

[56] McKinsey. Internet matters: The Net's sweeping impact on growth, jobs, and prosperity . `http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/internet_mat`
2011.

[57] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *Proc. of ICDT*, 2005.

[58] J. Mikians et al. Span-Dec data structure in portscan detection. Technical report. http://monitoring.ccaba.upc.edu/portscan/portscan-report.pdf, 2010.

[59] J. Mikians et al. Towards a statistical characterization of the interdomain traffic matrix. In *IFIP Networking*, 2012.

[60] Jakub Mikians. $heriff Browser Extension. http://pdexperiment.cba.upc.edu.

[61] Jakub Mikians, László Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. Detecting price and search discrimination on the internet. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, pages 79–84. ACM, 2012.

[62] S.Y. Nam, H.D. Kim, and H.S. Kim. Detector SherLOCK: Enhancing TRW with Bloom filters under memory and performance constraints. *Computer Networks*, 2008.

[63] A. Nucci, A. Sridharan, and N. Taft. The problem of synthetically generating ip traffic matrices: initial recommendations. *ACM SIGCOMM Computer Communication Review*, 2005.

[64] Andrew Odlyzko. Privacy, economics, and price discrimination on the internet. In *Proceedings of the 5th international conference on Electronic commerce*, pages 355–366. ACM, 2003.

[65] Andrew Odlyzko. The evolution of price discrimination in transportation and its implications for the internet. *Review of Network Economics*, 3(3), 2004.

[66] Andrew Odlyzko. Open Access, library and publisher competition, and the evolution of general commerce. http://www.dtc.umn.edu/~odlyzko/doc/libpubcomp.pdf, 2013.

[67] L. Olejnik, C. Castelluccia, and A. Janc. Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns. HotPETs '12.

[68] Eli Pariser. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin, 2011.

[69] V.n Paxson. Bro: a system for detecting network intruders in real-time. *Comput. Netw.*, 1999.

[70] S. Robertson et al. Surveillance detection in high bandwidth environments. In *Proc. of DARPA Information Survivability Conference and Exposition*, 2003.

[71] M. Roughan. Simplifying the synthesis of Internet traffic matrices. *ACM SIGCOMM CCR*, 2005.

[72] Schechter et al. Fast detection of scanning worm infections. In *Proc. of RAID*, 2004.

[73] J. Seibert et al. The Internet-wide Impact of P2P Traffic Localization on ISP Profitability. *IEEE/ACM Transactions on Networking*, 2012.

[74] S. Sen and J. Wang. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Transactions on Networking (ToN)*, 2004.

[75] A. Sridharan, T. Ye, and S. Bhattacharyya. Connectionless port scan detection on the backbone. In *Proc. of IPCCC*, 2006.

[76] Susan Davis. H.R. 6508: To direct the Federal Trade Commission to promulgate rules requiring an Internet merchant to disclose the use of a price-altering computer program, and for other purposes.

[77] The New York Times. Amazon's Prime Suspect. `http://www.nytimes.com/2010/08/08/magazine/08FOB-medium-t.html`.

[78] The Wall Street Journal. On Orbitz, Mac Users Steered to Pricier Hotels. `http://online.wsj.com/article/SB10001424052702304458604577488822667325882.html`.

[79] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon. Providing public intradomain traffic matrices to the research community. *ACM SIGCOMM Computer Communication Review*, 2006.

[80] University of Oregon Route Views Project. http://www.routeviews.org.

[81] H.R. Varian. Price Discrimination and Social Welfare. *The American Economic Review*, 75(4):870–875, 1985.

[82] N. Weaver, S. Staniford, and V. Paxson. Very fast containment of scanning worms. In *Proc. of the 13th Conf. on USENIX Security Symposium*, 2004.

[83] Wired. Online Prices Not Created Equal. `http://www.wired.com/techbiz/media/news/2000/09/38622.`, 2000.

[84] Tim Wu. Network neutrality, broadband discrimination. *J. on Telecomm. & High Tech. L.*, 2:141, 2003.

[85] Xinyu Xing, Wei Meng, Dan Doozan, Nick Feamster, Wenke Lee, and Alex C Snoeren. Exposing inconsistent web search results with bobble. In *Passive and Active Measurement*, pages 131–140. Springer, 2014.

[86] Y. Zhang et al. Fast accurate computation of large-scale IP traffic matrices from link loads. In *ACM SIGMETRICS*, 2003.

[87] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. *ACM SIGCOMM Computer Communication Review*, 2009.