DIPLOMARBEIT

Sampling DNS Traffic: A Day in the Life of the .at-Zone

ausgeführt am

Institut für
Stochastik und Wirtschaftsmathematik
TU Wien

unter der Anleitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Matthias Templ

durch

Andreas Blatt BSc. BSc.
Matrikelnummer: 00705817

Wien, am 17.08.2020

Supervisor:

Priv.-Doz. Dr.tech. Matthias Templ

Abstract

This thesis investigates the added utility of statistical sampling to DNS network traffic analysis, in particular with regards to issues of long-term storage and computation latency.

Using DNS log data for a full "day in the life of the Austrian Internet" provided by the Austrian domain registry operator nic.at, three emblematic sampling methods, namely *simple random sampling*, *systematic sampling* and *stratified random sampling*, are applied to a selection of network traffic features to assess their effectiveness in preserving the "true" population parameters.

Confirming theoretical considerations and previous research into Internet traffic, it was found that due to the query arrival process being highly self-similar, and also autocorrelated, systematic sampling leads to very precise estimates particularly for time-based traffic characteristics. For network traffic features independent of time, all sampling procedures in essence perform the same. Furthermore, it was shown that for tasks not involving very rare phenomena or the estimation of the number of distinct client IP addresses, sampling provides an easy way for fast data exploration with estimates for (frequent) traffic that are either practically identical to or less than 10% away from the true parameter (for patterns occuring at least on the same level as the sampling fraction) for the analysed features. Used in conjunction with current big data technology, these findings could lead to great gains in computation speeds and reduced storage requirements.

The method that consistently performed best or virtually indistinguishable from the others was systematic sampling, with the added benefit of being the computationally cheapest.

Acknowledgements

First and foremost, I want to thank my advisor Prof. Dr. Matthias Templ for his patience, trust and support - and not least for encouraging me to enroll at the Vienna University of Technology for my Master's degree this one evening in Spain after a day filled with R.
It's been quite a ride, but je ne regrette rien, and my fascination for computer-based Statistics and Mathematics so far is unbroken.
My gratitude also goes to my contacts from nic.at: Alexander Mayrhofer who got this whole ball rolling and introduced me to this thing that lies at the heart of the Internet called "DNS", Michael Braunöder, without whom the conversion of the data to a format I could work with would have taken at least double the amount of time and Otmar Lendl of cert.at for answering technical questions.
I also am endlessly grateful to my family for unconditionally supporting me in my ongoing quest for knowledge, for providing an oasis to come home to, an open ear anytime I needed it and the occasional eyeroll, even if I did not need that.
Of course at this point also a shoutout to my friends who tolerated my bursts of enthusiasm whenever I got to talking about my thesis, and were there for me when my head needed cooling. Once this gets printed, some typos and bumpy English will have vanished because of them, and for this I will buy you all $\epsilon > 0$[1] drinks. Finally, thank you Katharina for being the first person to read the full manuscript and saying nice things about it, even though it took up time I would have rather spent with you. Thank you for your support, encouragement, affection and for being in my life in general.

[1] $\epsilon \in \mathbb{N}, \epsilon << \infty$

Contents

List of Figures

List of Tables

1 Introduction

The Internet becomes ever more important to ever more aspects of daily life, with the number of devices connected to IP (abbreviation for *Internet Protocol*) networks being projected to be more than three times the global population by 2023 [Cisco Systems, Inc., 2018].
Looking up the next restaurant with at least four stars or more, finding a specific piece of information, a fridge automatically ordering milk or having a video conference with a colleague from the other end of the globe - at the bottom of all these applications lies the *Domain Name System* (DNS).
It is a hierarchical distributed database that can be thought of as the "address book" of the Internet. Specifically, it maps domain names like *www.example.com* to their IP address or back. The upper echelons of the hierarchy, the so-called *top-level domains* (TLD) like *.com*, *.org* or *.at* are managed by registry operators, like *nic.at* for Austria.
Registry operators maintain the list of domain addresses in their *zone* of the Internet and provide the infrastructure for users worldwide to access them. But their objective is not solely to manage the registration of domains and keep the servers running: An important aspect of their unique position in the Internet is traffic analysis and cybersecurity. For this purpose, *nic.at* logs all query activity in the *.at*-zone.
However, as traffic volume continues to grow, analyzing and storing the amount of traffic data puts an enormous strain on the used hardware to keep latencies close to realtime and storage requirements within manageable boundaries.

1.1 The Case for Sampling

Although recent years brought great progress in technologies for storing and analyzing "Big Data", especially with regards to the DNS [van Rijswijk-Deij et al., 2016, Wullink et al., 2016a,b], the use of sampling techniques is widespread in general network traffic analysis. Even today, it is a vivid area of research, even though its origins date back to the early 90s, see for example Amer and Cassel [1989], Claffy et al. [1993], Jedwab et al. [1992].

A random sample's attractive property for network traffic engineering is that it preserves the characteristics of its parent population to a degree that depends solely on its size, without requiring to know any of them in advance. Based upon a sample, it is possible to calculate aggregates that are only marginally different from the "true" value, or filter for specific components that are represented with a proportion close to their "true" incidence. Furthermore, it can significantly speed up network reporting tasks or computationally demanding analyses and provide fast glimpses into the structure of the data.

1.2 Thesis Contribution and Scope

The bulk of research on DNS traffic is directed towards the detection of botnets or malicious domains, with little focus on sampling, which is of course due to most researchers being limited to publicly available traces, trying to gather as much data as possible to accurately capture traffic characteristics. Domain registries on the other hand already have all the data, focussing their published research on Big Data solutions, such as SIDN's *ENTRADA* framework [Wullink et al., 2016a]. To the best of the author's knowledge, so far no research specifically concerned with the usefulness of sampling for DNS traffic analysis has been published.

The present thesis aims to close this gap, by a) investigating the applicability of sampling techniques for tasks domain registries are faced with by extending results from general network traffic analysis, and b) using this research to establish some baseline properties of (Austrian) DNS traffic for future researchers in analogy to work done by Castro et al. [2008] in their studies "A day at the Root of the Internet" and "Understanding and preparing for DNS evolution" [Castro et al., 2010]. This is made possible by the dataset provided by *nic.at*: A whole "day in the life of the Austrian Internet", which allows a rare look into the inner workings of DNS traffic from the vantage point of a *country-coded TLD* (ccTLD).

The analysis framework for this thesis is: Sampling occurs at ccTLD-level and ex post, as for security reasons in the everyday business situation the full log data has to be saved for a holdover time. This precludes the investigation of *adaptive* sampling methods that adapt their *sampling rate* to traffic load, as for example described by Choi and Zhang [2006], Choi et al., 2002a,b], Dogman et al. [2010], Drobisz and Christensen, Silva et al. [2013], since domain registries need to log all traffic in any case.

Two further approaches that will not be considered due to our particular research scope are *hash-based filtering* [Duffield, 2004, Molina et al., 2009] and *flow-based* sampling [Duffield, 2004, Duffield et al., 2002, 2005a,b]. A *flow* in this context means a set of DNS

queries with a common set of properties, like the same origin and destination [Quittek et al., 2004].
Hash-based filtering is a deterministic technique, where a packet or a flow gets selected on the basis of a hash computed from its characteristics.
Since in this thesis' scope sampling happens with the population still being available afterwards, sampling for flows is not necessary, as any flow of interest can be acquired through filtering. Hash-based selection thus would also only be a different way of filtering an already available dataset.
Therefore, this thesis' analysis focuses exclusively on the following notorious packet-based sampling methods: *n-out-of-N* resp. *Simple Random Sampling* (SRS), *Systematic Random Sampling* (SYS) and *Stratified Random Sampling* (STR) [Claffy et al., 1993, Molina et al., 2009], as they are the most fundamental approaches, and other random-based procedures ultimately inherit their useful properties from them.
In section 2.4.4, several characteristic traffic features such as the *query arrival process* at nic.at's name servers, the number of *unique IP addresses* or the *distribution of query types* are employed to compare the performance of the three methods in characterizing the *.at*-zone.

1.3 Outline

Chapter 2 poses the question of whether sampling is actually useful to domain registries, provides an introduction into the DNS system and establishes technical terms used later on in the thesis. Also, it discusses related work and how it informed the research undertaken. The chapter concludes with a discussion of the features selected for assessing the sampling approaches.
Chapter 3 first discusses the fundamentals of sampling theory and introduces SRS, SYS and STR as well as the requirements for good estimators. Moreover, rigorous conditions will be presented to assess the relative precision of each sampling method.
Chapter 4 introduces the dataset and discusses how the findings from chapter 3 translate to DNS traffic analysis. In parallel to applying the sampling methods and assessing their effectiveness, the selected features of Austrian DNS traffic will be comprehensively characterized. Furthermore, the choice of stratification variable as well as how bounds on precision for varying sample sizes can be determined will be addressed.
Chapter 5 summarises and evaluates the results obtained, presents the limits and value-added of sampling for DNS traffic analysis and proposes topics for further research.

2 Research Context

The primary goal of this thesis is to explore the usefulness of sampling techniques for domain registries managing a TLD, such as nic.at. This chapter explains the Domain Name System (DNS) and the structure of a query message.
Regarding sampling, the current trend goes towards *data streaming warehouses* [Wullink et al., 2016a] which are very capable at addressing several of the issues network operators have been struggling with since the early 90s, like data storage or realtime monitoring. Thus, the question is: *What value does the use of sampling add that is not already covered by current technologies?*

2.1 Advantages of Sampling

Phrasing the question as above creates the impression one can only use either "old-fashioned" sampling or "modern" data warehousing technology. But, as this thesis aims to show, using current warehousing technologies and established statistical concepts in unison can lead to a deeper understanding of hidden structures in the data or more expressive indicators. Even though advanced platforms such as ENTRADA take care of preprocessing and storing query logs, at some point there will be the question of *long-term storage*, or more specifically *what* to store.
According to Duffield [2004], with network traffic data the most common approaches to reduce data volume are:

- *Aggregation*, by combining several data points into a composite (like number of queries for a single domain), and storing only the composite. For example, a recent publication by Foremski et al. [2019] using DNS data collected at several globally distributed probes repeatedly aggregates the collected data, and after a holdover period stores only the longer aggregates, such as hourly, daily or weekly.
- *Filtering*, by selecting only data points with certain characteristic (like queries sharing similar properties) and discarding the rest.
- *Sampling*, by randomly selecting a proportion of the total.

It is evident, that with Aggregation and Filtering information and granularity is lost. With Sampling, that is of course also the case, but as will be argued in this thesis to a far lesser extent.
A sample of appropriate size preserves all information of its parent population, with the only price being a quantifiable loss of accuracy. Thus, a sample could still be filtered for patterns arising sufficiently frequently, aggregates calculated from the sample would still be reliable estimators for the population's "true" parameters and if the sampling method preserves the stochastic properties of the population, then the sample could also be used for the training of Machine Learning applications. Therefore sampling can actually be a tool to *preserve* information for the long term, while in parallel reducing storage requirements even further.
Besides the issue of storing data, another task where one can ask why to even bother with sampling is *DNS traffic analysis*.
Here, again the ENTRADA platform developed by SIDN, the Dutch domain registry, sets the standard. In Wullink et al. [2016a], the authors showcase ENTRADA's performance by running what they call *aggregation* and *scan queries*, with "scan" queries corresponding to filter-based selection in the above list, on a year's worth of DNS data. According to them, in a test environment the aggregation query took only 3.5 minutes to analyse 2.2 terabytes of DNS traffic stored in the native ENTRADA format[1], while the scan-type query took shortly over 3h.
Impressive as they are, these results arguably could be improved upon by employing sampling methods. Especially when it comes to calculating means, totals or proportions, sampling can be useful, since capturing such metrics accurately (enough) was the main reason why sampling was even conceived. If we assume a framework as was used by Wullink et al. [2016a], combined with using i.e. only 5% or less of the data, using a resource-efficient sampling method such as SYS, we expect even faster computation. Sampling could also be used to significantly speed up scan queries, of course with the caveat that, depending on sample size, "rare" patterns will slip through the filter. Still, we will show that using sampling methods will accurately preserve patterns such as the most frequently queried unique IP addresses or domains or shifts in the arrival of DNS messages.
Another application where sampling could prove a helpful addition, is with *statistical modelling* of DNS traffic. Through working with samples, it would become easier and

[1] In the course of preprocessing, ENTRADA converts the original *pcap*-files for captured DNS traffic to a file format more suited for analysis. Pre-conversion, the size of the dataset would have been 52 terabytes.

faster to fit and cross-validate parametric models for DNS traffic data, which would allow domain registries to draw inference going beyond the tracking of traffic attributes. This could be used for forecasting traffic loads, classification tasks or flagging irregularities in the query (inter-)arrival process.

2.2 The Domain Name System

Before discussing features for analysis and the current state of research into DNS traffic analysis, the Domain Name System will be introduced, in particular how it matches IP addresses to a domain name. Furthermore, the components of a DNS message will be described.

In the early days of the Internet, at that time called *ARPANET*, when the number of *hosts*[2] was still low in digits, every computer connected had a file called `HOSTS.TXT` which contained the names and IP addresses of all available hostnames. "Host" in this context means the specific computer in the network that provides the content someone might wish to access. The use of host "names" instead of simply using the hosts' IP addresses became necessary when the number of hosts became too large to memorize. The *Network Information Center* (NIC), a central authority of that time, had to approve each new name manually and add them to the register, making it necessary for network administrators to update their `HOSTS.TXT` regularly.

As the number of addresses grew further, it became increasingly difficult to keep the file updated and names unique, which in 1983 led to the development of the distributed hierarchical structure that the DNS has to this day ([Mockapetris, 1983a,b])[3].

2.2.1 The Domain Name Space

In the DNS, a domain name consists of five distinct components called *labels*, where each label corresponds to a level of the DNS tree shown in figure 2.1. Each level is administered by clusters of servers that are responsible for their particular label. They refer queries to the next-lower level, with information becoming more specific the farther down one gets on the tree. The labels are separated by dots and are processed from right to left, with the rightmost label being the *root node*, although the trailing dot is usually omitted.

The root level is followed by the *top-level domains* (TLD). TLDs can be either *generic*

[2]Name for a device connected to a network; for the DNS "hostname" can be used interchangeably with "domain name".

[3]For a more detailed history of the DNS, see for example Albitz and Liu [1997], Liska et al. [2016] or refer to the cited RFCs.

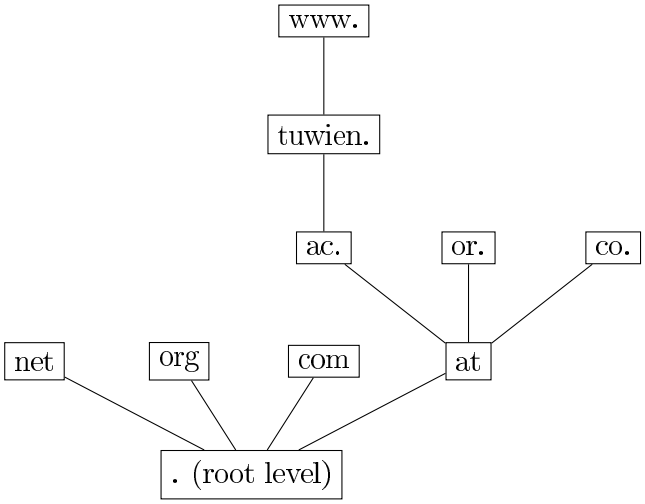


Figure 2.1: Schematic depiction of the DNS Tree

(gTLD), such as `.com.` or `.org.`, or *country-coded* (ccTLD) and are managed by domain registries like *nic.at* for the Austrian *zone* or *Verizon* for `.com`.
A DNS "zone" is a subset of the DNS managed by single administrator, who is responsible for maintaining the records of this segment. However, a zone need not represent a whole branch of the tree. The label after the TLD is referred to as *subdomain* or *second-level domain* (SLD) and can be followed by any number of subdomains.

2.2.2 Name Resolution

The primary purpose of the Domain Name System is to map domain names to their alphanumeric IP address, a process called *resolving*. The procedure for resolving an address is highly standardized and is sketched in the next paragraphs.
First, the host checks its local cache whether the requested IP address, i.e. for `www.example.com.` was resolved recently.
It is important to note that usually an IP address is only valid for a certain amount of time, its so-called *time-to-live* (TTL), after which the address has to be refreshed.
If the IP address isn't cached, the locally implemented resolver ("stub resolver") then queries a DNS resolver, typically a *recursive* DNS (RDNS) resolver run by the user's Internet Service Provider (ISP). Often, the process already stops here, as ISPs usually have the IP addresses of common domains ready.
Such an answer would then be called *non-authoritative*, as it didn't directly come from a *name server*. It is also not uncommon for ISPs and even browsers to prefetch IP addresses

for popular domains, or subdomains such as search results to reduce loading times[4].
If the ISP also does not have the IP address in cache, the next step taken by the RDNS resolver is to traverse the DNS hierarchy. For the domain `www.example.com.`, it starts with the rightmost label "." by contacting one of the 13 root servers[5].
The root server then directs the RDNS resolver to the IP address of a TLD name server in charge of ".com".
The RDNS queries the subsequent server for `www.example.com.`, which points it to another IP address of yet another DNS server responsible for the subdomain ".example". Once this recursive procedure reaches the server responsible for `www.example.com.`, the response the RDNS gets has the *authoritative* flag, as the server giving the answer is the *Authoritative Name Server* (AuthNS) for this domain. This AuthNS also sets the TTL for a DNS response. Finally, the RDNS returns the IP address to the user and, i.e. content from the site, can be retrieved.

2.2.3 Structure of DNS Messages

The messages sent from resolver to name server and back are called *query* and *response* and share the same basic structure: A *header* section followed by four sections with variable length: *question*, *answer*, *authoritative* and a section for *additional information*. The answer, authoritative and additional information sections all have the same format and are called *Resource Records* (RR). Table 2.1 presents the contents of a DNS message. Mockapetris [1987] specifies that both TCP (*Transmission Control Protocol*) and UDP (*User Datagram Protocol*) can be used as the underlying transport protocol for DNS messages, recommending the lightweight protocol UDP for queries and the more secure TCP for *zone transfers*[6], although regular hosts can just as well only use TCP. Mining the information like this contained in DNS messages is one of the principal tasks of DNS traffic analysis. As a domain registry can easily recreate the response to a given query, the following explanations and subsequent analyses of this thesis will focus solely on queries.
Besides the information listed in table 2.1, the header of a DNS query also contains the following fields:

- A timestamp, giving the time of arrival in microseconds,

[4] See for example: `https://www.chromium.org/developers/design-documents/dns-prefetching`, retrieved August 17, 2020
[5] `https://root-servers.org/`, retrieved August 17, 2020
[6] A zone transfer happens, when a subordinate server ("slave") requests the most recent zone file from a "master" server.

Section	Field	Description
Header	ID	Identifier assigned by the resolver that created the DNS query
	QR	flag if message is a query or a response
	OPCODE	type of query the message is carrying
	AA	flag if responding server was authoritative
	TC	flag if message was truncated due to its length
	RD	requests the receiving server to answer the query recursively
	RA	flag if responding server supports recursion
	Z	not in use currently, reserved for future use
	RCODE	response codes indicating if query was successful or not
	QDCOUNT	number of questions in the "Questions" section
	ANCOUNT	number of responses in the "Answer" section
	NSCOUNT	number of RRs in the "Authority" section
	ARCOUNT	number of RRs in the "Additional" section
Question	QNAME	requested domain name
	QTYPE	type of question being asked, specifies type of RR requested
	QCLASS	specifies class of RR requested
Resource Records	NAME	requested domain name
	TYPE	indicates format of the data, also indicates purpose of query
	CLASS	specifies class of RR requested, same as in "Question"
	TTL	number of seconds IP address is valid
	RDLENGTH	size of the data field
	RDATA	contains additional RR-specific data

Table 2.1: Structure of a DNS message

- the IP address of the host sending the query (*source IP address*) and the port from which the query was sent (*source port*),
- the IP address of the DNS server (*destination IP address*) and the port to which the query is sent (*destination port*) and
- the size of the DNS message.

The RRs make up what is called the *payload* of the DNS message and by analysing them it is possible to investigate traffic composition.
For example, if `TYPE = A`, this indicates that the query is asking to resolve a domain name to its IPv4 address (or its IPv6 address in the case of `TYPE = AAAA`), while `TYPE = MX` is typically used for a mail exchange.
`CLASS` identifies the protocol family used, although this field is practically always the class `IN` for "Internet". `TTL`, as was already mentioned, represents the time an IP address is

ID	Type	Function
1	A	maps domain name to IPv4 address
28	AAAA	maps domain name to IPv6 address
5	CNAME	maps domain name used as an alias to its *canonical name*
6	SOA	returns authoritative information about a DNS zone
2	NS	query for name of a zone's authoritative name server
12	PTR	used for reverse lookup of an IP address
15	MX	specifies which host handles e-mail for a domain
16	TXT	Text field with multiple uses
255	*/ANY	requests any/all records

Table 2.2: Common Resource Record Types

valid before it is automatically deleted from cache. There are many different RR types with some of the most common being shown in table 2.2, see Mockapetris [1987] for an extensive overview.

2.3 State of the art in DNS traffic analysis

As was discussed in the previous section, drawing random samples from the full data can reduce storage requirements without the same loss in granularity that come with aggregation or filtering, while estimates from the sample still provide similar, "precise enough" results for a range of traffic characteristics.

2.3.1 DNS traffic analysis

Research conducted on the DNS falls mainly in two major categories: The first one is concerned with detecting irregularities, botnets or malicious domains using DNS data, with Dodopoulos [2015] providing an extensive review into this field's State of the Art. The second area is focused on studying and mining DNS traffic characteristics using a wide range of data sources, ranging from root server and ccTLD traffic logs to traffic collected at the Internet links of campus networks.
A possible reason for sampling not being a primary concern in DNS research is the researcher's vantage point: Most researchers don't have access to ccTLD query logs and therefore gather as much data as they can to get a representative sample. On the other hand, works published by researchers associated with domain registries leverage their

full access and rather use "Big Data" technologies to bypass bottlenecks in computation than sampling methods.
Still, both areas are a valuable source for traffic analysis features that could profit from the application of sampling methods. The concluding section of this chapter will revisit research on DNS traffic for feature extraction.

2.3.2 Sampling Network Traffic

When it comes to sampling, one important point to consider is at which vantage point the sample is drawn: In the DNS context, domain registries are not troubled by too little data, as it comes their way in any case, and researchers not associated with a domain registry try to collect as much as possible.
In the related field of *general* network traffic analysis though, the use of sampling methods is not only a subject receiving lots of attention, but also a necessity. While DNS servers do get a lot of traffic, the volumes hitting Internet Service Providers (ISP) are larger by orders of magnitude as they have to handle all non-DNS related traffic as well. But similar to domain registries, ISPs and other network operators are interested in monitoring their network for performance, usage statistics, trends or anomalous behaviour as close to realtime as possible to meet service level demands, plan their resources optimally and store data for "post-mortem" analyses. The most common approach is *Packet Sampling*, where a random sample is drawn out of the stream of packages arriving e.g. at a server.
Research into using packet sampling for network measurements dates back to the beginning of the 90s, with seminal work done by Amer and Cassel [1989], Jedwab et al. [1992] and Claffy et al. [1993].
Amer and Cassel proposed the use of sampling to allow close-to-realtime monitoring of network traffic characteristics, specifically with regards to phases of peak traffic when the calculation of statistics would take up too many resources at the measurement point. Also, they provided a first classification of sampling methods, which correspond to *Simple Random Sampling* (SRS) and *Systematic Random Samling* (SYS) in this thesis' framework. They concluded their paper describing a theoretical approach to estimate the mean packet size and detecting structural breaks via a statistical test for difference in means.
Jedwab and Phaal, also arguing under the assumption of constrained processing and storing resources, discussed how sampling can be used to get sufficiently accurate packet arrival counts, by focussing only on the t largest source-destination pairs.
The arguably most influential work in the field of packet sampling for network traffic

classification was published by Claffy et al. [1993]. They compare the performance of *event*- and *time*-driven sampling methods corresponding to SRS, STR and SYS in characterizing network traffic data collected at the entrance point into an Internet backbone network. To assess the degree of accuracy, they used a χ^2-based goodness-of-fit test to compare the sampled distributions of packet sizes and packet interarrival times to the respective distributions of the parent population. This approach was chosen because many network traffic characteristics come from skewed or multimodal distributions where the mean is not a useful metric.

FOK [2003] discusses the application of stratified sampling to network traffic, showing that stratification can lead not only to increased precision but also to smaller required sample sizes, suggesting the use of packet sizes as a stratification variable. This will be investigated in estimating the query arrival process in chapter 4.

The previously listed research, in combination with findings on filtering, trajectory-sampling and hash-based procedures, were consolidated and standardized into a *Request for Comments* (RFC), a non-binding recommendation for network operators, by the *Packet Sampling* (PSAMP) working group of the *Internet Engineering Task Force* (IETF) in Molina et al. [2009].

In the decades that followed, research into sampling network traffic diversified into several different strains focussing on different aspects of the initial problem of how to deal with large data volumes and how to analyse network traffic in a timely fashion.

Two very active branches are *adaptive sampling* approaches that focus on ways to dynamically alter the sampling rate according to traffic load, and *flow-based sampling* approaches which aim to more accurately capture spatial movement in the network than packet-based trajectory procedures, by specifically sampling packages that share common properties. As was discussed in the chapter 1, these topics are not in the scope of this thesis.

For an overview into research applying various versions of packet sampling to different fields of network traffic analysis see for example Silva et al. [2017], and Duffield et al. [2005b] for a discussion of flow-based sampling.

This thesis will for the most part follow the methodology proposed by Claffy et al. [1993], as the author regards their approach as the most fundamental, but will place a stronger emphasis on the mathematics behind the reason for the relative performance of the sampling methods.

The notion of using goodness-of-fit tests to measure a sampling method's precision will not be pursued due to the different scope of this thesis. One important traffic engineering task is monitoring query behaviour *over time*, and since Internet traffic strongly depends

on human biorhythm, using distribution tests runs the risk of disregarding the temporal structure of DNS traffic.
Furthermore, analysis will focus on event-driven sampling, as time-based sampling was found to perform worse overall by Claffy et. al., which is very likely due to the *burstiness* of Internet traffic described by Leland et al. [1994] where traffic spikes have no natural "length". This means, time-based sampling in particular could lead to a larger loss of information, since traffic spikes might just as likely happen before or after the sampling window, as interarrival times are not exponentially distributed.

2.4 Feature Selection

The fields of a DNS message described above allow for a variety of analyses. To show that many of them can just as well be computed by using a fraction of the data with only a minor loss in accuracy, this section will present and group features frequently used in literature on DNS research. Beside the detection of anomalous patterns, many of the features discussed in the following paragraphs are also used in general DNS traffic research using the "Day in the Life of the Internet" dataset, where data from the root level of the DNS is collected and examined, see Castro et al. [2008, 2010].

2.4.1 Temporal Features

Characteristics such as the number of queries arriving or the number of unique IP addresses per timeframe, changes in types of RRs requested or shifts in countries of query origin can provide helpful insights for network management tasks and be leveraged for the detection of anomalies, botnets or malicious domains.
By investigating traffic over time at two university DNS servers, Zdrnja [2006] found that automated spam detection software contacting servers hosting information on IP addresses of known spam sources creates a constant and non-negligible flow of `TYPE = A` queries. Also, they found that a notable proportion of `TYPE = TXT` queries can be attributed to email traffic due to the *Sender Policy Framework*[7], which uses `TXT` records to identify legitimate email servers.
Other examples for how the temporal structure of DNS traffic can be leveraged to detect abnormal or malicious patterns range from investigating changes in the covariance structure ([Zheng and Shyong, 2011]), tracking the relation between the number of unique IP addresses and their respective query counts over time ([Yuchi et al., 2010]) or focussing

[7] See Wong and Schlitt [2006].

on domain names with abnormally high or temporally concentrated queries ([Villamarin-Salomon and Brustoloni, 2008]), to cite just a few out of a very active field.
Another measurement that could be used is the distribution of interarrival times, the time that passes between two queries arriving at a name server. As observed by Leland et al. [1994], this distribution is likely heavy-tailed. By having a model for "regular" interarrival-times, this might be used to detect denial-of-service (DOS) attacks.

2.4.2 Geolocation Features

To investigate the geographic origin of the logged queries, the *MaxMind GeoIP*[8] database was used in this thesis to match countries of origin to the source IP addresses of the queries in the available dataset. This enables analysis of time zone related patterns, with Bilge et al. [2011], Stalmans et al. [2012] discussing how the geographic location of servers can be used to detect botnets. Data on cities of origin would also have been available, but due to a large number of missing values, this level of granularity was not pursued.

2.4.3 DNS Query-based Features

By investigating the different fields of a DNS query (table 2.1), further features for analysis can be derived. The timestamp from the header section plays an important role for time-based analyses, while the number of unique source IP addresses can be used to detect malicious flux networks, networks where one domain name typically maps to many distinct but only fleeting IP addresses (Bilge et al. [2011], Perdisci et al. [2012]). The TTL-field also allows much inference to be drawn, since Bilge et al. [2011], Mahjoub et al. [2014], Perdisci et al. [2009] found that malicious domains typically have a TTL shorter than 150 seconds. Furthermore, the TTL-field can be used to identify the operating system of the client that sent the query.
As was discussed above, tracking the composition of RRs allows a detailled view of DNS traffic composition for anomalous behaviour ([Zdrnja, 2006]), but has also more "mundane" applications in general network management.
In particular, the distribution of query types can be used to track the adoption of IPv6 or the degree of source port randomization [Castro et al., 2008, 2010], with the latter being important due to a vulnerability of the DNS protocol found by Dan Kaminsky [Team, 2008]. This attack type is commonly called "cache poisoning" and takes advantage of DNS cache resolvers frequently not using source port randomization. Another frequent

[8] `https://www.maxmind.com/en/home`

topic in DNS anomaly detection is directly mining the queried domain names, as the number of different characters, dots or the length of the domain name can be indicative of phishing domains or botnets as is for example investigated by Antonakakis et al. [2010], Bilge et al. [2011], Fette et al. [2007], Yadav et al. [2010], Yarochkin et al. [2013], Zhang et al. [2007].

2.4.4 Features for Analysis

For any sampling method to be of use to domain registries, a sample drawn under its design must provide sufficiently accurate estimates of network traffic characteristics and, if required, their distribution over time.
Since this thesis ultimately cannot possibly account for all thinkable analyses, or reproduce the above mentioned topics with the available dataset[9], it is necessary to impose some restrictions.
For example, an interesting topic although not commonly associated with DNS traffic would be to analyse the relationship between query arrival and the interarrival times.
But as will be shown later, DNS traffic is highly non-stationary which makes investigating this question ultimately a topic future research or a thesis of its own.
The presented sampling methods, albeit naturally less so for stratified sampling, are rather unconcerned with the particular content of a query message, that is, it is inconsequential whether the characteristic one is interested in estimating is traffic's resource record composition, the country of query origin or the transport protocol used.
Thus, provided that the "degree of randomness" is similar for different features, the results that are reported later are transferable to different, but similar features. Furthermore, if there was a statistically significant shift or a time-dependent extreme pattern, it can be expected that it will be reflected in any random sample for any method, although to varying degree. This will be the subject of the following chapters.
Table 2.3 consolidates the previous discussion into the features that will be used to assess the performance of SRS, SYS and STR in this thesis, and whether they will be estimated over time or as daily aggregates.

[9] For another, more practically oriented overview into further analysis topics see `https://entrada.sidnlabs.nl/query_examples/`, retrieved August 17, 2020

Type	Feature
time-based	Number of queries arriving per minute
	Traffic decomposition by frequently requested resource records per minute
	Number of active resolvers per minute
aggregation and filter-based	Daily traffic decomposition by frequently requested resource records
	Daily number of active resolvers
	Queries for most frequently requested google domains
	Query counts by source countries
	Adoption of IPv6
	Distribution of TTLs

Table 2.3: Features analysed in this thesis

3 Methods

The need for selecting samples from a population to draw inference initially arose from *censuses*[1] being prohibitively expensive or even impossible. Naturally, the question then is how to best sample and how reliable the results from the used samling method, or *survey design* are[2]. Survey designs can be classified in two general categories, particularly *probability* and *nonprobability* sampling.

In probability sampling, each element of the population has a "known", nonzero probability of being included in the sample, while nonprobability designs select the elements based on some other criteria, like quotas, ease of access or a-priori assumptions.

In some situations, where even probability sampling would be hard or expensive to implement, nonprobability sampling can be a useful substitute, for example in opinion polls or market research.

The main advantage of probability based sampling methods though is that their *validity*, *reliability* and *precision*[3] can be assessed in advance with sampling theory before even drawing a sample.

By contrast to the usual sampling situation, this thesis faces the inverse case where the whole population is available, except for queries answered through caching. Due to the privileged position of domain registries in the network, a full census happens every day.

In previous chapters, the DNS and related research were introduced. Now, the next step is to show how and to what extent sampling provides reliable estimates of traffic characteristics and why it preserves a parent population's characteristics better than aggregation or filtering. To this end, subsequently some important results from sampling theory are presented.

The next sections will introduce and establish important properties of SRS, SYS and STR, discuss how their relative performance can be measured and highlight the circumstances under which they differ.

[1] A census is a full survey of the population.

[2] "Sampling method" and "Survey design" can be used interchangeably. For a history of science on random sampling methods, the reader is referred to D.R. Bellhouse's introductory chapter in Krishnaiah and Rao [1988].

[3] The concepts and notation used throughout this chapter follow Levy and Lemeshow [2009] and Cochran [1977], unless otherwise noted.

3.1 Sampling Fundamentals

The (target) *population* N represents the total set from which a *sample* of size n is selected according to some method. The population is made up of sampling *units* $i \in 1, ..., N$ that each have a characteristic θ_i which can take on a variable *value* of X_i.
For DNS traffic, the population could represent traffic arriving at a name server, with queries being the sampling units and the log files in the order of their arrival the *sampling frame*.
Recalling table 2.1, each of the fields can be a characteristic with which to estimate population *parameters*, like *totals* (X), *means* ($\bar{X}$) or *proportions* ($\frac{X}{N}$) and their corresponding variances. Population parameters will subsequently always be denoted in capital letters. If n units are randomly selected from the population, sample analogues to the population parameters can be calculated which are called *estimates* and will be denoted by by lowercase letters. The *estimators* for parameters and the corresponding sample estimates are shown in table 3.1.
In sampling theory, the default method of selecting a sample is what in this thesis will be referred to as *simple random sampling* (SRS).
To estimate population totals $\hat{X}$, the estimate x is multiplied by $\frac{N}{n}$, the inverse of the *sampling fraction* denoted by f. The sampling fraction gives the size of the sample relative to the population, while the inverse is commonly called the *inflation* or *expansion* factor.
In opposition to the idealized setting of sampling theory, where a sample is drawn from an infinite population, surveyors in a practical setting are typically concerned with large, but ultimately finite populations. Therefore, when calculating the population variance with an estimate, the *finite population correction* (fpc) $\frac{N-n}{N}$ has to be introduced when the sample contains a significant proportion of the population to correct for the artificial precision of a larger sample. It is evident, that for small n, the fpc remains close to unity. According to Cochran [1977, pp. 25], up until sampling fractions of 5%, or in some cases even 10%, the inclusion of the fpc can be omitted.

3.2 Properties of Estimators

As was discussed in the introduction to this chapter, the advantage of probability sampling is that the validity, reliability and precision of the estimators resulting from a sampling method can be quantified up front. These three qualities are closely related to the following fundamental mathematical criteria of a "good" estimator.

3.2.1 Unbiasedness

The *bias* of an estimate is defined by the following relation.

$$B(\hat{X}) = \mathbb{E}(\hat{X}) - X.$$

The concept of "bias" is used to measure systematic errors in the sampling procedure, and is one of the reasons why probability sampling is preferable to nonprobability methods: Even a large sample could result in wrong estimates if the selection procedure was not randomized, as it likely misrepresents the true frequency of characteristics. For example, it was found by Claffy et al. [1993] that time-based sampling of network traffic resulted in less accurate estimates than event-based sampling.
One possible reason, related to bias, could be the bursty nature of Internet traffic: If all observations of the a random 10-minute window of an hour were sampled, this would result in a large and superficially even representative sample, but it would consistently over- or underestimate traffic, depending on whether the patterns observed in the sampling windwo of an hour are actually representative for the whole time slot.
Conversely, an estimator $\hat{x}$ is called "unbiased", if its expected value is equal to the true parameter in the population.

$$\mathbb{E}(\hat{X}) = X.$$

The statistical law hiding in plain sight, that leads to this result is the *Central Limit Theorem* (CLT). Formulated for our purposes[4], the CLT reads as follows:

Theorem 3.2.1 (Classic Central Limit Theorem)**.** Let $\bar{X}$ be a population mean and $x_1, ..., x_n$ the i.i.d. characteristics observed in a random sample of size n with mean $\mathbb{E}(x_i) = \bar{x}$ and variance $s^2 < \infty$. Then

$$\sqrt{n}\frac{(\bar{X} - \bar{x})}{s} \xrightarrow{d} \mathcal{N}(0, 1), \tag{3.1}$$

or

$$\bar{x} \sim \mathcal{N}(\bar{X}, \frac{s^2}{n}). \tag{3.2}$$

The appeal of the CLT as stated above is that especially for large sample sizes n, the mean, and by extension the total (see table 3.1), estimate of an estimator $\bar{x}$ are normally distributed, with the expected value being the true parameter. This property will enable

[4] Adapted from Theorem 2.4.1 in Lehmann [2004, pp. 73].

us to compute approximate confidence bounds for our estimates later on.
For a review of the concept *convergence in distribution* used in the above theorem or proof, the reader is referred to standard textbooks such as Billingsley [2012], Lehmann [2004].
A sampling method that exhibits only limited or zero bias will even on repetition with the same data always provide *valid* estimates.

3.2.2 Efficiency and Measures of Precision

As a measure of quality, unbiasedness is often found to be insufficient, as it is relatively easy to construct estimators that *on average* result in unbiased estimates. The common criterion to assess the *efficiency* of estimators is comparing the dispersion of their estimates around the true parameter. The "best" unbiased estimator thus would have the lowest amount of dispersion, which means that the estimates resulting from it are not only unbiased, but also on average closer to the true parameter.

Definition 3.2.1 (General Form for a Measure of Precision)**.**

$$\begin{aligned} MP(\hat{X}) &= \mathbb{E}(\phi(X - \hat{X})) \\ &= \frac{1}{n}\sum_{i=1}^{n} \phi(X_i - \hat{X}_i). \end{aligned} \tag{3.3}$$

The above definition gives the mean dispersal of the estimates from the true parameter with regards to some function ϕ.
Typical choices for ϕ are the L^2 norm which corresponds to the *mean square error* (MSE), and the L^1 norm which corresponds to the *mean absolute error* (MAE). The MAE and after taking the root, the root MSE (RMSE) both are on the same scale as the data measured. It is often practical to scale error measures by the true parameters to get scale-free measures of precision.
By comparison, due to the squaring of the errors, the RMSE puts more weight on extreme differences and is harder to interpret than the MAE. Where the latter simply calculates the mean of the absolute differences between the sample estimate and the true parameter, the RMSE is the *mean value of squared deviations about the true value of the parameter being estimated* Levy and Lemeshow [2009, pp. 35]. Hyndman and Koehler [2006] critically compare these and other measures of accuracy in greater detail.
Statistically, the MSE has the useful property of being directly related to the variance of the estimator, which can be shown with some elementary algebra and is therefore ideally

suited for the theoretical comparison of methods.

Definition 3.2.2 (Mean Square Error)**.**

$$\begin{aligned} MSE(\hat{X}) &= \mathbb{E}((X - \hat{X})^2) \\ &= \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2 \\ &= Var(\hat{X}) + Bias(\hat{X})^2 \end{aligned} \tag{3.4}$$

So, if the estimator associated with a sampling method is known to be unbiased, the MSE reduces to the sample variance, and the estimator with the lowest associated variance then would provide the most *precise* estimates. An interesting implication of 3.2.2 is that estimators with a small, but known bias can result in more precise estimates than unbiased estimators if they have small variation. For the reporting of the results in this thesis though, where the actual population is available to gauge the sample-based estimation, the MA(P)E[5] will be used, as it is a scale-free measure and allows for easily interpretable measures of validity.

Definition 3.2.3 (Mean Absolute Percentage Error)**.**

$$MAPE(\hat{X}) = 100\frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - \hat{X}_i}{X_i}\right| \tag{3.5}$$

3.2.3 Consistency

A requirement even more fundamental than bias and efficiency is an estimator's *consistency*. An estimator is called *consistent* if it becomes increasingly more precise for increasing sample size. So, for an unbiased and efficient estimator, if $n \to N$, the MSE should be approaching zero.
Formally, the consistency of an estimator can be derived directly from the following formulation of the *Law of Large Numbers*[6].

Theorem 3.2.2 (Weak Law of Large Numbers)**.** Let $\bar{X}$ be a population mean, and $x_1, ..., x_n$ the i.i.d. characteristics observed in a random sample of size n with mean

[5] MAPE: Mean Absolute Percentage Error, the MAE scaled by the true value.
[6] Adapted from Theorem 2.1.2 in [Lehmann, 2004, pp. 49]

$\mathbb{E}(x_i) = \bar{x}$ and variance $s^2 < \infty$. Then the sample average $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ satisfies

$$\bar{x} \xrightarrow{P} \bar{X}, \tag{3.6}$$

or

$$\lim_{n \to N} \mathbb{P}(|\bar{X} - \bar{x}| > \epsilon) = 0 \quad \text{for any } \epsilon > 0. \tag{3.7}$$

What this theorem states, is that the *probability* for an arbitrarily large difference between the true population and the sample's estimate goes to zero as the sample size increases. In combination with 3.2.1 this ensures the *reliability* of a specific sampling method, as even on repeated sampling with different sample sizes the estimate will be quantifiably close to the true parameter. For a review of the concept *convergence in probability* used in the above theorem or a proof, the reader is referred to standard textbooks such as Billingsley [2012], Lehmann [2004].

3.3 Simple Random Sampling

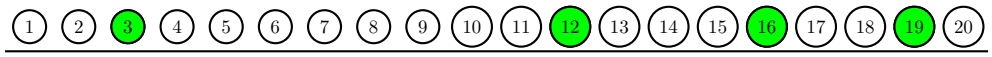


Figure 3.1: Schematic depiction of *Simple Random Sampling*

SRS is the most fundamental probability-based way to draw a sample of size n out of a population of size N without replacement. The way SRS will be characterized in this thesis corresponds to *content-independent* and *n-out-of-N* random sampling according to PSAMP's categorization of sampling methods for network traffic ([Molina et al., 2009]), as package content will not be relevant in the sample selection. Furthermore, sampling will be based on generating n random numbers in the range of $[1, N]$ and then selecting the queries with the corresponding index position.
By sampling n out of N, the population is separated into $\binom{N}{n}$ possible samples, with each sample having the same probability of $1/\binom{N}{n}$ of being selected. Now the previously discussed properties for the SRS estimators shown in table 3.1 can be investigated. For this, exemplary results for the mean and variance estimates will be presented, specifically their unbiasedness and the variation of the estimates themselves[7].

[7]The theorems presented in this section are found in Cochran [see 1977, Chapter 2]

Statistic	Population	Estimator
Total	$X = \sum_{i=1}^{N} X_i$	$\hat{x} = (\frac{N}{n}) \sum_{i=1}^{n} x_i$
Mean	$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$	$\bar{x} = \frac{1}{n} \sum_{i=1}^{N} x_i$
Proportion	$P_X = \frac{X}{N}$	$p_X = \frac{x}{n}$
Variance	$\sigma_X^2 = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{X})^2$ for proportions: $\sigma_X^2 = P_X(1 - P_X)$	$s_x^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$ $s_x^2 = \frac{np_x(1-p_x)}{(n-1)}$ or $s_x^2 = p_x(1 - p_x)$ for large samples

Table 3.1: Population Parameters and SRS Estimators

3.3.1 Bias of the Estimators

Theorem 3.3.1. The average of the sample mean $\bar{x}$ over all possible SRS samples is an unbiased estimate of $\bar{X}$.

$$\mathbb{E}(\bar{x}) = \sum_{i=1}^{n} \frac{x_i}{\binom{N}{n}} \tag{3.8}$$

Theorem 3.3.2. For a SRS sample, s_x^2 taken over all $\binom{N}{n}$ possible samples is an unbiased estimate of σ_X^2:

$$\mathbb{E}(s_x^2) = \sigma_X^2. \tag{3.9}$$

Proof. For proofs to these theorems, see Cochran [1977]. □

3.3.2 Variance of the Estimators

Theorem 3.3.3. The variance of a SRS mean estimate $\bar{x}$ over all possible samples is given by

$$\begin{aligned} Var(\bar{x}) &= \mathbb{E}((\bar{x} - \bar{X})^2) \\ &= \frac{s^2}{n} \frac{N-n}{N} \\ &= \frac{s^2}{n}(1 - f), \end{aligned} \tag{3.10}$$

where f denotes the sampling fraction $\frac{n}{N}$.

Corollary. The standard error of $\bar{x}$ over all possible samples is given by

$$\begin{aligned} SD(\bar{x}) &= \frac{s_x}{\sqrt{n}}\sqrt{\frac{(N-n)}{N}} \\ &= \frac{s_x}{\sqrt{n}}\sqrt{(1-f)}. \end{aligned} \tag{3.11}$$

Proof. For proofs to these theorems, see Cochran [1977]. □

Corollary 3.3.2 shows, that the standard deviation of the mean estimate is directly proportional to the variance estimate of the characteristics x_i in the population and inversely proportional to the square root of the sample size. Thus, it can be inferred that 1) increasing sample size increases precision, but with diminishing returns, and 2) estimates for a highly variable characteristic require larger sample sizes. Characteristics with little variation of course require significantly smaller sample sizes.
In theorem 3.3.3 the finite population correction enters the equation when the condition of sampling from a finite population N is added.

3.4 Stratified Random Sampling

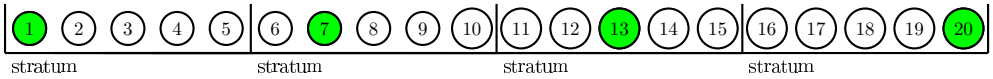


Figure 3.2: Schematic depiction of (proportional) *Stratified Random Sampling*

STR is a natural extension of simple random sampling to increase precision. Instead of drawing a random sample from the total population to survey a characteristic, it can be practical to divide or *stratify* the population into into L non-overlapping subpopulations (*strata*) $N_h, h = 1, ..., L$ according to a *stratification* variable, such that

$$N = N_1 + N_2 + ... + N_L.$$

The stratification variable is either correlated to the characteristic of interest, or even better, the characteristic itself. From these strata then samples of size n_h are drawn by some probability-based method, frequently but not necessarily SRS.
This makes STR a composite technique as per the categorization of Molina et al. [2009]. Furthermore, STR is a *content-dependent* technique as, in the context of DNS traffic analysis, it requires inspection of query content.

Statistic	within Stratum	Population Estimator
Total	$\hat{x}_h = \frac{N_h}{n_h} \sum_{i=1}^{n_h} x_{hi}$	$\hat{x}_{str} = \sum_{h=1}^{L} \hat{x}_h$
Mean	$\bar{x}_h = \frac{1}{n_h} \sum_{i=1}^{n_h} x_{hi}$	$\bar{x}_{str} = \frac{1}{N} \sum_{h=1}^{L} N_h \bar{x}_h$
Proportion	$p_{x_h} = \frac{x_h}{n_h}$	$p_{x_h,str} = \frac{1}{N} \sum_{h=1}^{L} p_{x_h}$
Variance	$s_{hx}^2 = \frac{1}{n_h - 1} \sum_{i=1}^{nh} (x_{hi} - \bar{x}_h)^2$ for proportions per stratum: analogous to table 3.1	only stratum specific estimates relevant for STR

Table 3.2: STR Estimators: within strata and population estimates

Stratification can result in great gains in precision, depending on the choice of stratification variable and the degree of its relation to the characteristic of interest.
This is also unfortunately one of the major downsides of STR since it requires prior knowledge about 1) the relative sizes of the strata for correct allocation of strata sample sizes and 2) about whether a stratification variable is correlated to a characteristic of interest. In the context of DNS traffic analysis it generally involves a greater computational overhead, as all these considerations have to be taken into account before computing a stratified estimate.
The gains in precision are especially great, if stratification divides the population into subgroups that are internally more homogenous than the total.
This thesis for example, albeit mainly for computational reasons, stratified Austrian DNS traffic by the geographic location of nic.at's name servers, which likely increases the precision of estimates for the proportions of queries coming from specific countries and more accurately reflects the proportion of traffic that is handled by each name server.
If a simple random sample is drawn from each stratum, then the estimators for each stratum inherit the previously discussed properties of the SRS procedure.

It is important to note, that the STR estimate for the mean $\bar{x}_{str}$ is not necessarily equal to the sample mean $\bar{x}$ from SRS, as the means for each stratum are weighted by the factor $W_h = \frac{N_h}{N}$, the so-called *stratum weight*.
Thus, the SRS and STR estimates for the mean coincide only when the sampling fraction is the same in all strata, which is called a *proportional allocation* of the n_h. STR samples resulting from proportional allocation are also called *self-weighting* samples.
In sampling situations where a full census is not available, often other methods of allocating the respective strata sizes are used, like *equal allocation*, where all strata have the same size, or *optimal allocation*, where the size of each stratum sample is dependent on its variation. Optimum allocation can lead to reduced sample size requirements while

also maintaining a high precision, and equal allocation can be useful for statistical tests of differences between strata.
As the features analysed in this thesis (see table 2.3) in one way or another represent proportions of the parent population, STR uses proportional allocation for maximum accuracy in the frequency estimates.
In the following theorems, again exemplary for the mean, taken from [Cochran, 1977, pp. 91ff], it can be seen that STR also provides unbiased results for the mean and the variance, even if the underlying procedure was not necessarily a simple random sample.

3.4.1 Bias of the Estimators

Theorem 3.4.1. If the estimate for the sample mean $\bar{x}_h$ in every stratum is unbiased, then $\bar{x}_{str}$ is an unbiased estimate of the population mean $\bar{X}$.

$$\begin{aligned} \mathbb{E}(\bar{x}_{str}) &= \mathbb{E}\frac{1}{N}\sum_{h=1}^{L} N_h \bar{x}_h \\ &= \mathbb{E}\sum_{h=1}^{L} W_h \bar{x}_h \\ &= \sum_{h=1}^{L} W_h \bar{X}_h \end{aligned} \tag{3.12}$$

Proof. The above theorem can be proven easily by rewriting the definition for the population mean and taking into account the expected value.

□

Theorem 3.4.1 highlights the strength of STR: As long as an independent sampling procedure is applied to each stratum results in unbiased stratum estimates, STR's population estimate will also be unbiased.

3.4.2 Variance of the Estimators

Theorem 3.4.2. The variance of the sample mean estimate $\bar{x}_{str}$, provided sampling in different strata happened independently from each other, is given by

$$Var(\bar{x}_{str}) = \sum_{h=1}^{L} W_h^2 Var(\bar{x}_h), \tag{3.13}$$

where $Var(\bar{x}_h)$ is the variance of the stratum specific mean estimate over all possible samples.

Proof. To prove the above theorem, we recall a basic property of the variance for linear functions:

$$\begin{aligned} Var(\bar{X}) &= Var(\sum_{h=1}^{L} W_h \bar{x}_h) \\ &= \sum_{h=1}^{L}\sum_{j=1}^{L} W_h W_j Cov(\bar{x}_h, \bar{x}_j) \\ &= \sum_{h=1}^{L} W_h^2 Var(\bar{x_h}) + 2\sum_{h=1}^{L}\sum_{j>h}^{L} W_h W_j Cov(\bar{x}_h, \bar{x}_j). \end{aligned} \tag{3.14}$$

Since the samples were selected independently in all strata, the covariance terms are zero and the theorem is proven. □

Theorem 3.4.2 highlights the importance of the stratum weights in getting an accurate estimate for a population characteristic: The more stratification reduces within-stratum variation, the more precise a STR will become.
So far, no constraints were put on the procedure generating samples from each stratum besides independence – that means sampling in one stratum does not influence sampling in another, and the resulting estimates being unbiased.
If this underlying procedure is now assumed to be SRS, theorem 3.4.2 can be reformulated to account for this additional information.

Theorem 3.4.3. The variance of the sample mean estimate $\bar{x}_{str}$, provided sampling in different strata happened independently via SRS, is given by

$$\begin{aligned} Var(\bar{x}_{str}) &= \frac{1}{N^2}\sum_{h=1}^{L} N_h(N_h - n_h)\frac{s_h^2}{n_h} \\ &= \sum_{h=1}^{L} W_h^2 \frac{s_h^2}{n_h}\left(\frac{N_h - n_h}{N_h}\right) \\ &= \sum_{h=1}^{L} W_h^2 \frac{s_h^2}{n_h}(1 - f_h). \end{aligned} \tag{3.15}$$

Proof. This result can by proved by applying theorem 3.3.2 to an individual stratum and substituting into theorem 3.4.2. □

Remark. If the sampling fractions f_h are negligible ($f < 5\%$) in each stratum, the above

variance formula reduces to

$$Var(\bar{x}_{str}) = \sum_{h=1}^{L} W_h^2 \frac{s_h^2}{n_h}. \tag{3.16}$$

3.4.3 Relative Precision of STR and SRS

Using the respective variances for the mean estimate, it is possible to assess the efficiency of STR and SRS. Of course it should be noted, that STR not automatically results in smaller variance for estimates since its performance rests entirely on the choice of stratification variable.
If the number of queries arriving per minute were to be estimated and stratified by query message size, which is a property that does not vary much with time due to the high degree of standardisation of a DNS message, the resulting estimate would functionally be the same as an estimate computed by SRS. On the other hand, if the goal was to accurately reflect the distribution of query message size, selecting a proportional random sample from each "bucket" of queries with equal size will on average greatly outperform a SRS estimate.

Theorem 3.4.4. If the finite population correction can be disregarded and strata sample sizes are allocated proportionally ($f_h = f$), the following relationship from theorems 3.3.3 and 3.4.3 can be deduced:

$$Var(\bar{x}_{str}) \leq Var(\bar{x}_{srs}) \tag{3.17}$$

Proof. For the proof, see Cochran [1977, pp. 99ff]. The general idea is to decompose the population variance into a general variation and variation within strata part and successively replacing terms with the variance of mean estimates for SRS and STR, resulting in the following equation:

$$Var(\bar{x}_{srs}) = Var(\bar{x}_{str}) + \frac{1}{n} \sum W_h (\bar{X}_h - \bar{X})^2 \tag{3.18}$$

□

Therefore it can be concluded, that the increase in mean precision for STR comes from the elimination of the within-stratum variances that strongly depend on the choice of stratification variable. If on the other hand stratification does not increase homogeneity within strata, there is on average no difference between STR and SRS estimates.
In circumstances of non-negligible sampling fractions, it is even possible for proportional

STR to perform worse than SRS, particularly when heterogeneity within strata is larger than the similarity of strata with each other.

3.5 Systematic Sampling

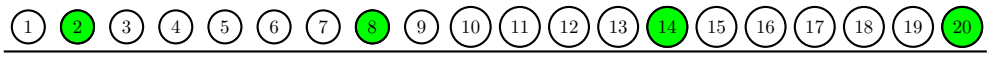


Figure 3.3: Schematic depiction of (count-based) *Systematic Random Sampling*

While SRS is conceptually the easiest sampling method, this does not necessarily translate to practice. Especially the condition of knowing the inclusion probability of any unit in the population is very restrictive. In an ex-post situation as considered in this thesis, where the total number of queries for any time period is only a matter of finding the last row in a database table, this downside is only a minor issue, but there is an even simpler way to select a sample with considerably less overhead, namely *Systematic Sampling*.
In this thesis, SYS will be implemented as follows: Instead of using random numbers for all index positions of observations selected, only one random number k from $[1, \frac{N}{n}]$ to is drawn to determine the *sampling interval* and then sample all observations whose index is a multiple of k.
For example, if a 1% sample ($\frac{N}{n} = 100$) were to be selected and the initial k was 36, then the next units would have index position 72, 108, 144,... and so on. One advantage of SYS here is that neither N nor n need to be known beforehand, only their desired approximate integer relation. Furthermore, this leads to a more even "spread" of the sample through the population.
Following the categorization of Molina et al. [2009], this corresponds to content-independent *count*-based systematic sampling.
It should be noted, that the size N of the population is not usually an integer multiple of k, which especially for small populations can result in samples of unequal sizes. This raises questions concerning the unbiasedness of results from SYS and the supposed equal probability of inclusion for each sampled observation, but as the population to be sampled here is very large – one might even speak of Big Data – these concerns can easily be ignored. Therefore, for the remainder of this thesis, it will be assumed that $N = nk$.
Conceptually, this rule corresponds to dividing the population into k possible systematic samples of size n and randomly choosing one, which makes SYS a case of *cluster sampling*.
The SYS estimators for population parameters are the same as those for SRS listed in

table 3.1, except that $\frac{N}{n}$ can be replaced by the chosen sampling interval. Also, provided the sampling interval is an integer or can be considered as one due to large N, the estimates are again unbiased.
The main difference between SYS and SRS lies in the variance of their estimators.

3.5.1 Variance of the Estimators

Theorem 3.5.1. The variance of a SYS mean estimate $\bar{x}$ over all possible samples is given by

$$Var(\bar{x}_{sys}) = \frac{N-1}{N}s^2 - \frac{k(n-1)}{N}s^2_{ws}, \tag{3.19}$$

where s^2_{ws}, the *variance among units within the same systematic sample*, is given by

$$s^2_{ws} = \frac{1}{k(n-1)}\sum_{i=1}^{k}\sum_{j=1}^{n}(x_{ij} - x_{i.})^2$$

Remark. Theorem 3.5.1 can also be stated in terms of the *intraclass* correlation coefficient between units in the same systematic sample.

$$Var(\bar{x}_{sys}) = \frac{N-1}{N}\frac{s^2}{n}[1 + (n-1)\rho_{ws}] \tag{3.20}$$

Proof. The proof for the above theorems can be found in [pp. 208ff]cochran77. □

Theorem 3.5.1 highlights an interesting contrast between SYS and STR: For SYS, the precision of an estimate actually is *increased*, the more heterogenous a systematic sample is, while STR's deviation from the "true" parameter decreases with more homogeneity within strata.

3.5.2 Relative Precision of SYS and SRS

From theorem 3.5.1 it is very easy to establish when a SYS sample yields more precise estimates than SRS. Similar to the comparison of STR and SRS, the variances of the mean estimates are now compared.

Theorem 3.5.2. A mean estimate resulting from SYS is more precise than a SRS mean estimate iff

$$S^2_{ws} > S^2. \tag{3.21}$$

Proof. Substituting and rearranging the below relation results in the above statement.

$$\begin{aligned} Var(\bar{x}_{srs}) &> Var(\bar{x}_{sys}) \\ (\frac{N-n}{N})\frac{S^2}{n} &> (\frac{N-1}{N})S^2 - \frac{k(n-1)}{N}S_{ws}^2 \end{aligned} \tag{3.22}$$

□

Thus, for SYS to be more precise than SRS, the variance within each systematic sample should be larger than the variance of the population. If SYS increases homogeneity, for example such that the intraclass correlation coefficient becomes positive, SYS becomes less precise as the amount of information that is captured by larger variation is lost. If the intraclass correlation is negative and low, which reflects a great amount of heterogeneity, SYS can in some cases even outperform STR at significantly reduced overhead. This is especially the case if the population is ordered with regards to the characteristic being measured, as would be the case for time-based features where both STR and SYS were found to perform particularly well.
In the case of ρ_{ws} being zero for no correlation between sampling units, as would be the case if the sampling units were ordered randomly, SYS estimates are on average the same as would result from SRS. This property was proven for averages over all finite populations by Madow and Madow [1944] and by postulating an infinite superpopulation and the average over all finite populations that can be drawn from them by [Cochran, 1977, pp. 213].
This result establishes that there is functionally no difference between SYS and SRS if the characteristic surveyed has no underlying order.
Therefeore, if determining the inclusion probabilities for SRS is not easily done or computationally infeasible, a SYS sample will typically yield equally as precise estimates.

3.5.3 Relative Precision of SYS and STR

In order to compare whether SYS or STR perform better for a certain survey feature, it is necessary to look more closely at how the population is structured. The precision of an SRS estimate is not affected by trends in the population or whether it is ordered by some characteristic, whereas this severly affects SYS and STR.
Madow and Madow [1944] conducted the first thorough study of the relative properties of STR and SYS, establishing some fundamental results: In the presence of linear trends, STR and SYS can both outperform SRS, but STR in general results in more precise estimates. In the case of periodic patterns in the population, SYS was found to be

more precise than STR if the sampling interval k is an odd multiple of the half-period, with STR resulting in better estimates when k is an integer multiple of the period. For autocorrelated populations, e.g. populations where observations are more similar the closer they are to each other, Cochran [1946] established an important relationship for finite populations assumed to be selected at random from an infinite superpopulation with a convex correlation function.

Theorem 3.5.3. Let observations $x_i, i \in 1, ..., N$ that are drawn at random from a superpopulation fullfill

- $\mathfrak{E}x_i = \bar{x}$
- $\mathfrak{E}(x_i - \bar{x})(x_{i+u} - \bar{x}) = \rho_u \sigma^2$
- $\mathfrak{E}(x_i - \bar{x})^2 = \sigma^2$,

where $\mathfrak{E}$ denotes averages over all finite populations that can be drawn from this superpopulation. Furthermore, let

$$\rho_u \geq \rho_v \geq 0 \quad (u < v)$$

and

$$\rho_u^2 = \rho_{u+1} - 2\rho_u + \rho_{u-1} \geq 0 \quad [u = 2, 3, ..., (kn - 2)].$$

Then, for any size of sample the subsequent equation is true:

$$\mathfrak{E}(Var(\bar{x}_{sys})) \leq \mathfrak{E}(Var(\bar{x}_{str})) \leq \mathfrak{E}(Var(\bar{x}_{srs})). \tag{3.23}$$

Furthermore, unless $\rho_u^2 = 0, u = 2, 3, ..., (kn - 2)$, SYS is always more precise than STR.

Proof. The above theorem is proven in Cochran [1946].

□

The above theorem has very deep implications for this thesis' analysis. As shown by Leland et al. [1994], Internet traffic, especially when viewed over time, is highly *self-similar*, which in essence corresponds to observations being strongly autocorrelated over time. Therefore, from this foray into the theoretical foundations of sampling theory, the following conclusions can be stated:

Ordering of Characteristics: If it can be assumed or established that the ordering of the population is random, and there is no obvious stratification variable available that can increase within-stratum homogeneity, SRS, STR and SYS result in estimates of equal precision. In such cases, it might be feasible to use SYS, as it is the method with the least computational overhead.

Correlation Structure: In autocorrelated populations, SYS is strictly more precise than STR, although in the some cases STR estimates can be of equal precision to SYS, for example if the within-stratum correlation reduces to zero. Another situation where proportionally allocated STR will likely perform equally as good or better than SYS is if the proportion of each stratum relative to the total is of interest, as the allocation ensures that the proportion will be very close to the population parameter.

4 Sampling DNS Traffic

In the chapters leading up to this, the presented material was focused on laying the groundwork for the following experiments. Now, that the DNS and the structure of a query message as well as the theoretical foundations for comparing sampling methods have been discussed, this chapter will turn to the practical application of these foundations.
After describing the dataset and the conducted preprocessing steps, the sampling methods described in chapter 3 will be applied to the features discussed in section 2.4.4 (see also table 2.3).
The resulting measures of precision will be provided after an exploratory data analysis for each feature, as the previous chapter found that the structure of the population to be sampled has great influence on the performance of the different sampling methods. Furthermore, the effect of various stratification variables on the STR estimates will be considered, as well as ways to construct approximate confidence intervals.

4.1 Dataset

The data provided by nic.at consists of all logged DNS traffic within the `.at`-zone on May 6th 2015 separated according to the geographical location of the server instances that manage the .at-zone worldwide.
It was made available in the *pcap* format commonly used for storing network traffic, with separate folders containing files spanning ten minute intervals for all locations where nic.at operates a name server. Before conversion to long format, the raw pcap-files took up approximately 50GB of disc space. Each file contains the resource records (RR) for both the queries arriving at the name server and the name server's response. For the subsequent analysis, only the records for full day available were considered.
These files were combined with *tshark*[1], fields were selected with *packetQ*[2], converted to long format and uploaded to a SQL database via a batch script for the statistical analysis

[1] `https://www.wireshark.org/`, retrieved August 17, 2020
[2] `https://github.com/DNS-OARC/PacketQ`, retrieved August 17, 2020

and further processing with *R* (R Core Team [2020]). It must be noted for the results reported later, as already mentioned in section 3.4, that for computational reasons the data were "stratified" by distinct name server instance at a geographic location, meaning that all sampling methods were run separately for each name server made available. The collected samples were then recombined and the respective features were estimated.
As a name server's response to any given query can be reproduced from its own RR, only queries were consideredfor analysis. Using the *MaxMind GeoIP*[3] database, information on the queries' country of origin was added. Besides the information on the country of query origin, each column corresponds to a specific part of the DNS message, with table 2.1 giving an overview on the fields available for analysis.
It is important to underline, that it is not possible to map an IP address to an individual user using this data without also having access to data from their respective ISP, as all traffic that arrives at nic.at's servers passes through an ISP first. Also, the ISPs generally hold their own cache with commonly requested domain names which they renew in regular intervals, which also masks the actual traffic volume to some extent.
In central hubs, such as the United States or in Austria, nic.at employs multiple servers working in parallel at the same location to better manage high-volume traffic. For initial exploratory analysis and hypothesis-building concerning the expected performance of sampling procedures, a small Vienna-based name server was used. The resulting training data set contained 4.519.528 queries, while the full dataset consisted of 259.362.849 queries for a day in Austrian Internet. Unless otherwise noted, the results reported will be for the *full* dataset.

4.2 Time-based Features

4.2.1 Query Arrival Process

In the review of research on DNS traffic in section 2.4.4, one of the most important features was found to be the rate at which queries arrive at a name server. This feature lies at the heart of most network management tasks and is also a very useful indicator of irregular behaviour. Any sampling method therefore must be able to provide precise estimates for the contour and the number of queries per time frame. In this thesis, only minute aggregation will be discussed in detail, as an estimate that is valid at this level will necessarily also be valid at higher time resolutions. The subsequent analysis was also conducted for second resolution, but other than requiring significantly larger

[3] `https://www.maxmind.com/en/home`, retrieved August 17, 2020

samples overall to reach feasible levels of precision, the relative performance of SRS, SYS and STR remained unchanged.

Population Characteristics

As was discussed in chapter 3, the structure of the population can already indicate which survey design will provide the most precise estimates.
For this feature, the query content is irrelevant, as only the proportion of total traffic in each arrival time slot is of interest. In essence, this means that the population parameters to be estimated are non-randomly *ordered* with regards to time of arrival which already portends SYS performing rather well.
Due to the high volume of DNS traffic arriving per minute, concerns that SYS' sampling interval could coincide with the arrival process' periodicity can safely be dismissed, except for sampling fractions ranging in the order of millions.

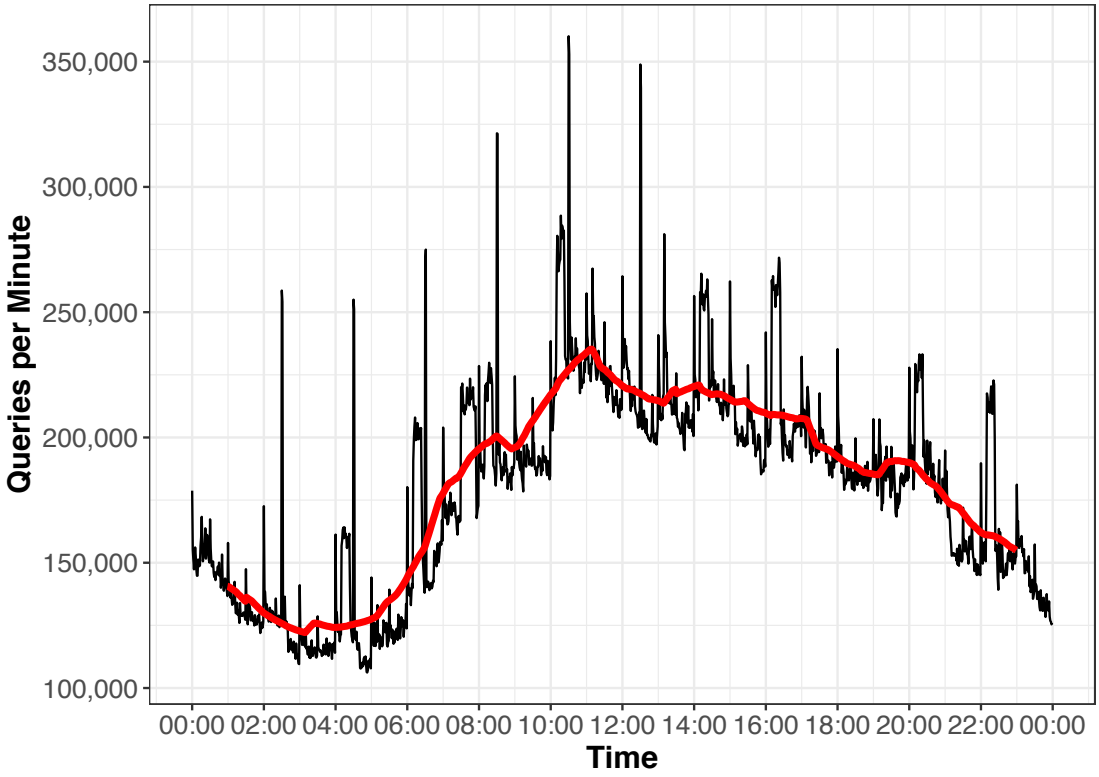


Figure 4.1: Query Arrival Process for the `.at`-Zone

Figure 4.1 shows the query arrival rate for the `.at`-zone on May 6th 2015.

The curve displays a diurnal pattern, which is typical for network traffic primarily driven by humans. This is particularly visible in the red 3h moving average, with low amounts of traffic during night time, growing traffic in the morning as more people wake up and check their emails and a marked "lunch dip" after 11 o'clock. Afterwards, traffic slowly declines back to the low volumes of nighttime.
Still, the curve shows some very distinct structural breaks with spikes that appear at very even intervals and periods, where traffic as a whole shifts upwards.
Interestingly, in these shifts, the apparent distribution of queries arriving does not so much change, as that it gets scaled up. Also, on closer inspection, there is always a recognizable spike in traffic at the beginning of each hour and the half hour mark. These might be indicative of some automated routines and might be of interest to network analysts or automated systems monitoring traffic.

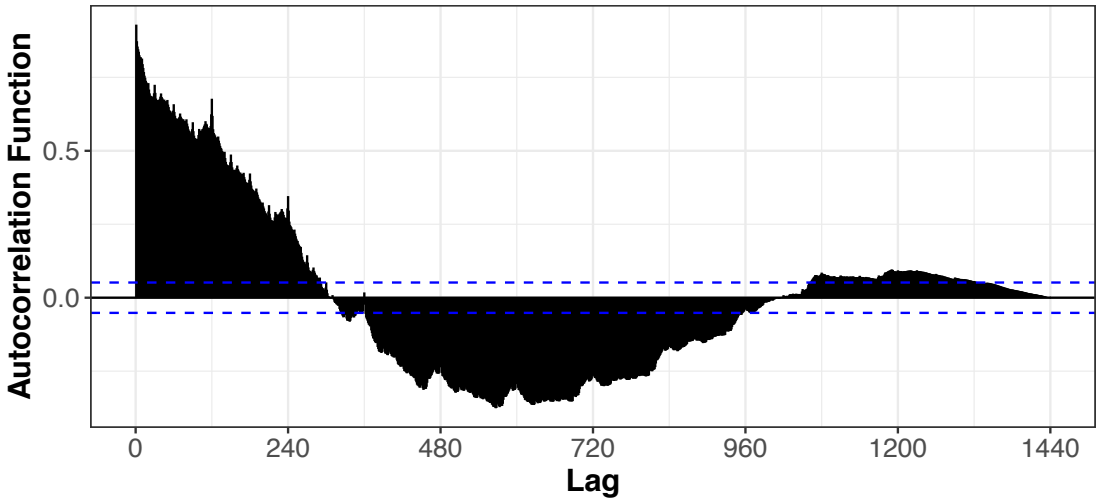


Figure 4.2: Autocorrelation Function for the Arrival Process

In figure 4.2, the query arrival process' autocorrelation function (ACF) is displayed. Here, the non-stationary and autocorrelated nature of the query arrival process is particularly visible. The ACF declines very slowly until lag 300 (corresponds to 5 a.m.) with the query arrival process between 6 a.m. and 4 p.m. being strongly correlated with each other, but negatively so with the initial hours of the day. The decline towards the lower nightly traffic volumes starting around 5 p.m. (lag 1020) again shows a positive correlation with the late night decline starting at midnight. This can be considered as indicative of different time regimes where DNS traffic behaves differently and shifts in

the underlying distribution of query arrivals occur.
Taking a cue from Leland et al. [1994], the query arrival process was aggregated to different time resolutions such as seconds or hours. The contour of the query arrival process, as well as the shape of the ACF did not change, which points toward *self-similarity*. A measure commonly used to measure self-similarity is the *Hurst coefficient*. It was calculated using a wavelet transform as described by Beran et al. [2013], and was estimated to be very close to 1. Therefore, it is evident that DNS traffic is strongly autocorrelated and highly self-similar.
This means that the behaviour of DNS traffic is preserved regardless of scaling in time, and that bursts in query arrival are very likely to be followed by another. For an in-depth overview into theory and research on self-similarity and long-range dependence in the context of Internet traffic modelling, the reader is referred to Karagiannis et al. [2004].
Recalling theorem 3.5.3 and the importance of autocorrelation for the relative performance of the sampling methods, SYS can therefore be expected to be more precise than either STR or SRS, unless the chosen stratification variable allows a more even spread of the sample over time than SYS.

Effects of Stratification

The set of possible stratification variables for this feature is limited, as the prime driving force of the pattern observed in figure 4.1 is the diurnal pattern resulting from human biorhythm.
It is therefore highly unlikely for any query field not directly related to the time of arrival (see table 2.2) to sufficiently decrease the within-stratum covariance.

This can be seen in figure 4.3, where the different sampling methods' precision is given as a function of sample size, including several possible stratification variables.
The inverse square root relationship discussed in corollary 3.3.2 is clearly observable in all curves, but the difference in precision between SYS, STR based upon strata for each minute (1440 bins for a whole day) and other considered stratification variables is remarkable.
Both SYS and STR require very small sample sizes for a highly precise estimate of traffic. Also, it is interesting to see that hourly stratification actually performs worse than SRS. A possible explanation for this can be the substantial variation per hour of the day seen in figure 4.1, in particular the number of bursts and machine-driven patterns on top of traffic being non-stationary.

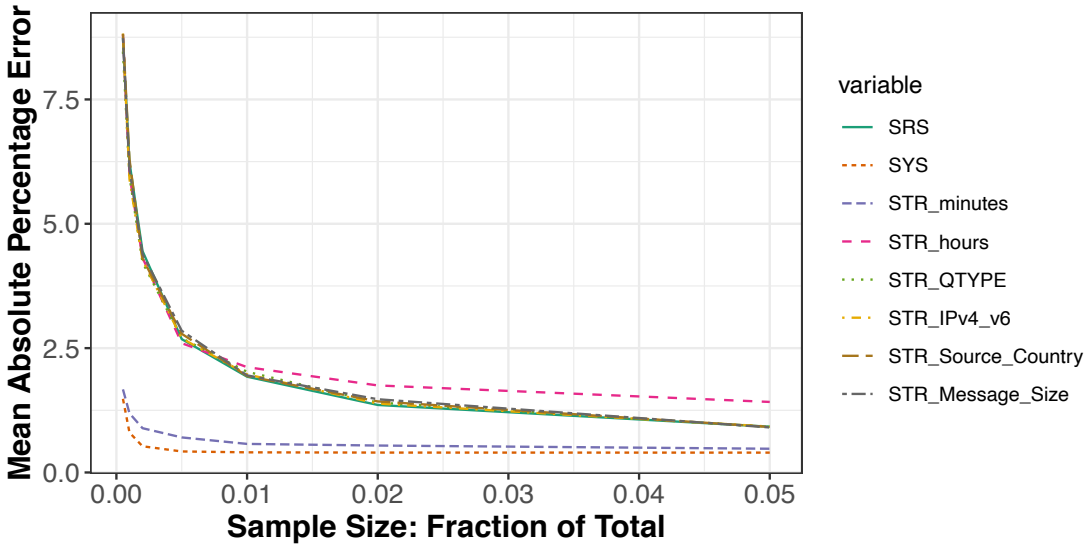


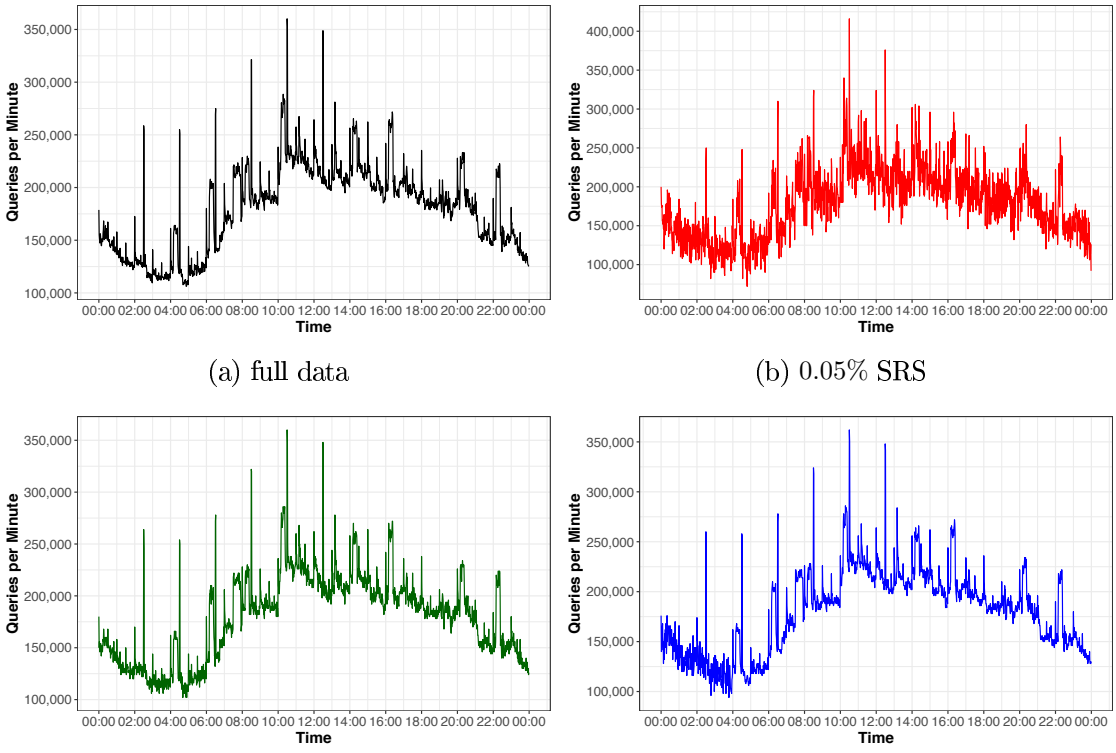
Figure 4.3: Precision as a Function of Sample Size and Sampling Method (Query Arrival)

Another way to read figure 4.3 is as a "test" of correlation, since only a stratification variable strongly correlated with the characteristic measured results in increased precision.
Thus, i.e. the requested resource record type, the Internet protocol version used or country of query origin can very likely be considered random attributes that don't vary significantly over time. The packet size, which was found to lead to increased precision for a different type of network traffic by FOK [2003], does also not appear to be correlated with the time of query arrival, which is likely due to the highly standardized DNS message format.
Again it must be noted that the employed analysis framework was already "stratified", i.e. drew samples separately for each name server instance at a geographic location for faster computation. As ISPs typically route traffic to the physically closer name server, this approach might already have leveraged gains in precision.[4].

Results

Figure 4.4 juxtaposes an exemplary graphical representation using 0.05% samples for each sampling method with the true contour. From the figures, it is apparent, that each method performs rather well in preserving the overall contour of daily traffic, especially

[4]For a discussion of the effect of *anycast*, where (root) name server instances share an IP address but are physically different nodes and how this affects routing see Liu et al..



(a) full data

(b) 0.05% SRS

(c) 0.05% SYS

(d) 0.05% STR (min)

Figure 4.4: Query arrival process, results of sampling

when it comes to the timing and height of the spikes. But, as can be seen in figures 4.4c and 4.4d, SYS and STR show markedly less variation in the estimates than SRS even at this low sampling fraction.

In order to better assess each sampling method's reliability, for each combination of sampling fraction and method 20 samples were drawn and the *mean absolute percentage errors*[5] was calculated. Table 4.1 summarizes the results of this procedure and gives the average MAPEs. The sampling fractions indicate what percentage of the total dataset consisting of over 250 million rows was used.

The actual sample sizes might differ slightly from the approximate numbers reported here due to the full data not necessarily being integer divisible.

The table and the figures confirms the assertion from theorem 3.5.3, since using SYS to estimate the daily query arrival process results by far in the most precise estimate. It outperforms even STR using the best stratification available in the dataset, which was

[5]See section 3.2.2 for a definition.

Sampling Fraction	Sample Size	SRS	SYS	STR (minute based)
0.0005%	129.158	8.65	1.40	1.72
0.001%	258.315	6.04	0.70	1.06
0.002%	518.630	4.29	0.34	0.67
0.005	1.291.576	2.71	0.14	0.37
0.01	2.583.152	1.91	0.07	0.25
0.02	5.166.304	1.34	0.03	0.17
0.05	12.915.761	0.84	0.01	0.10

Table 4.1: Relative Precision of Sampling Methods in Mean Absolute Percentage Errors for Queries per Minute

drawing a proportional sample from each minute bin.
A likely explanation for SYS outperforming STR here lies in their very design. SYS agnostically selects every k-th multiple of the randomized starting index, while proportional STR requires more information to correctly draw a sample. Since especially several smaller name servers show particularly low traffic during the late-night period from midnight to 4 a.m., with query counts often smaller or equal to the sampling fraction, it becomes difficult to draw a proportional sample of the desired size. This can result either in a sample where certain minutes have no observations, or a sample where certain bins that "should" have a fractional number of observations get rounded up to 1. Both cases lead to the distortions visible in 4.4d.
Ultimately, it needs to be pointed out, that drawing a time-based proportional stratified sample was by far the computationally most elaborate to implement, notably since the traffic engineering features are interested in accurate estimates of the counts per bin, which makes proportional sampling a necessity. Still, using SYS and only one-tenth of a percent (0.001%) of the full data already results in an error of less than 1%, which means that if the true arrival, i.e. at 5:36 a.m. were 100.000 queries, the sample's estimate would be off by less than 1000.

Confidence Limits

Given the size of the samples, the Central Limit Theorem (theorem 3.2.1) can be used to construct approximate confidence intervals for the estimates. In this case, as all methods were shown to be unbiased, the MAPE can be considered a measure of variation.
Therefore, constructing a confidence interval essentially amounts to multiplying the

MAPEs reported above with a quantile of the standard normal distribution, e.g. in the case of the 97.5% z-quantile, the MAPEs would be multiplied by 1.96.
Thus, the confidence bound for SYS estimates using 1% of the full data would still be less than one percent – if 100 samples were drawn from the population using SYS, only 5 would have a deviation from the true parameter larger than 1%.

4.2.2 Traffic decomposition by resource record types per minute

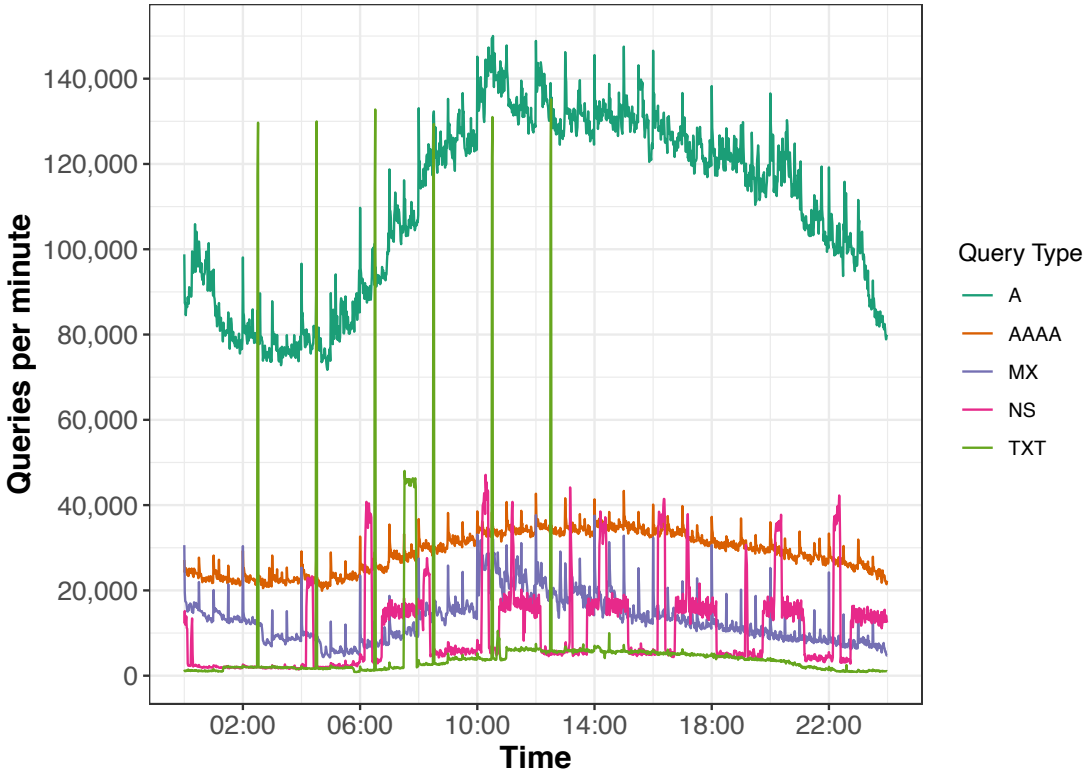


Figure 4.5: Traffic decomposition: Frequently requested requested RR types per minute

Monitoring the query arrival process provides valuable insight into whether DNS traffic is "normal" or if something noteworthy occured, such as the the regularly spaced spikes visible in figure 4.1.
What cannot be seen from solely investigating the number of queries arriving though, is where structural breaks likely originate or *what* the queries actually request, like the IPv4 or IPv6 address of a host (types `A` and `AAAA`) or host information for a mail exchange

(MX)[6].
A possible way to find out is to decompose traffic by resource record type and investigate changes in its components. Consequently, sampling methods applied to captured DNS traffic must be able to accurately reflect the query type proportions, especially for the most frequently requested.

Population Characteristics

Figure 4.5 displays five of the most frequent query types (missing for a complete top 7 selection are TYPE = ANY and TYPE = DS) and their development over time. The most frequently requested resource record by far is the IPv4 address of a host (TYPE = A), with requests for an IPv6 address (TYPE = AAAA) in a trailing second place.
Queries such as these translate host names like *www.tuwien.ac.at* to the IP address where the site can be found. Although IPv4 was officially replaced as Internet Standard by IPv6 in 2017 due to its limited address space (see Deering and Hinden [1998, 2017]), it is apparent that it is still up and about.
It also can be seen that the spikes and a lot of the variation visible in figure 4.1 result especially from bursts in NS and TXT records, with the former requesting information about an authoritative name server and the latter possibly being related to the *Sender Policy Framework* [Wong and Schlitt, 2006] and email authentification. The regularity and intensity present in the plot might be indicative of something else at work though.
Although rather well-behaved by comparison, queries for MX records also show regular spikes, especially at intervals of an hour. See e.g. Mockapetris [1987] for a more extensive discussion of resource record types. Table 4.2 lists the top 7 RR types and their proportion of total traffic.

Effects of Stratification

Although the extent differs for each type, decomposed traffic also in general follows the diurnal pattern established in the previous section.
Provided requests for different resource record types are not in more ways related to the time of day except for more people being awake requesting them, this feature can be viewed as a random attribute of a query that appears in no particular order.
Therefore, the three methods should show similar performances in precision as was found by the end of chapter 3, unless some stratification variable can be found that sufficiently increases within-sample homogeneity.

[6] See table 2.2 for a list of common resource record types

TYPE	Frequency	Proportion
A	158952661	0.605
AAAA	41720900	0.159
MX	18876168	0.0717
NS	14943446	0.0569
DS	8403419	0.0319
ANY	7410372	0.0282
TXT	7345880	0.0279

Table 4.2: Top 7 most frequently requested resource record types

In the following section, exemplary results for STR are shown using two different stratification variables.
Namely, in the first case proportional samples from a) minute bins and b) a proportional sample from each query type category are drawn.

Results

Figure 4.6 visualizes SYS' precision in capturing the daily contour of the five most frequent query types exemplary using a 1% sample. Even though sampling increases variation, it very accurately captures the timing and the height of the spikes, with only minor differences between figures 4.5 and 4.6.

Table 4.3 shows the mean measures of precision for 20 repetitions for each combination of sample size and method, underlining the assessment from above. SYS and time-based STR outperform SRS and `TYPE`-based stratification only in the most frequent category, which indicates that specifically for `TYPE=A` queries there could be a time-dependent pattern. But as hypothesized above, overall in general, and particularly at sampling fractions greater than 1%, there is virtually no difference between the methods.
The low precision for the more infrequent types is strongly dependent on sample size and the actual frequency of the population's characteristic in question.
Recalling table 4.2, the total proportion of `TXT`-type queries to total traffic amounts to around 2.8%. So to reduce uncertainty for this frequency at minute level, rather large sampling fractions would be required.

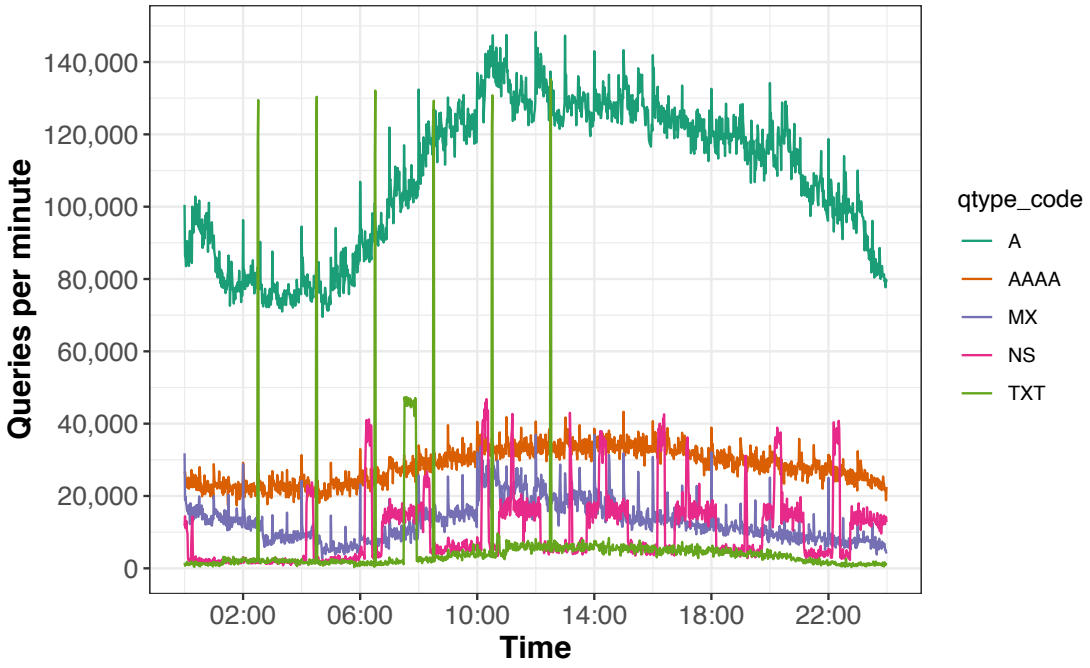


Figure 4.6: Sampled decomposition of query arrival into most frequently requested resource record types per minute (SRS, 1%)

Total proportion of traffic for frequent resource record types

On the other hand, if the time dimension is of no interest and only estimates for the *daily* proportions as listed in table 4.2 are required, sampling proves to provide remarkably accurate results, regardless of method or sampling fraction. This is also indicative of resource record type essentially being a randomly ordered attribute, although SYS and time-based STR reach better precision particularly for the query types that were responsible for the spikes and structural breaks in figure 4.5.
Table 4.4 summarises the relative precisions achieved by each method and sampling fraction for the resource record types analysed above.

One takeaway of table 4.4, again showing the mean precision for 20 repetitions, is that when it comes to calculating summary statistics of randomly ordered DNS attributes, choosing a sampling method becomes a question of preference and computational overhead, even though time-based stratification and stratification by `QTYPE` itself led to more precise results especially at lower sampling fractions. As will be seen for the subsequent features, the higher the time aggregation, the more precise summary estimates calculated

Sampling Fraction	A	AAAA	MX	NS	TXT
0.0005	11.04	21.05	32.51	39.34	58.92
0.001	7.86	15.15	23.56	30.93	42.65
0.002	5.77	10.76	16.78	23.26	32.57
0.005	3.90	6.91	10.61	14.96	21.88
0.01	3.00	5.05	7.63	10.61	15.52
0.02	2.48	3.73	5.45	7.55	10.91
0.05	2.12	2.67	3.61	4.97	6.87

(a) MAPEs for SRS

Sampling Fraction	A	AAAA	MX	NS	TXT
0.0005	6.94	19.27	31.32	37.44	57.49
0.001	5.01	13.75	22.58	29.35	41.50
0.002	3.77	9.70	15.97	22.38	32.36
0.005	2.73	6.32	10.19	14.13	21.55
0.01	2.31	4.60	7.31	10.08	15.15
0.02	2.10	3.47	5.21	7.13	10.72
0.05	1.99	2.52	3.45	4.68	6.69

(b) MAPEs for SYS

Sampling Fraction	A	AAAA	MX	NS	TXT
0.0005	7.23	19.30	31.32	37.49	56.84
0.001	5.42	13.78	22.72	29.21	41.06
0.002	4.08	9.90	16.02	22.16	32.04
0.005	3.03	6.54	10.24	14.33	21.44
0.01	2.58	4.84	7.34	10.17	15.18
0.02	2.33	3.69	5.35	7.42	10.79
0.05	2.19	2.70	3.68	4.93	6.82

(c) MAPEs for STR, (proportional) stratification by minute bins

Sampling Fraction	A	AAAA	MX	NS	TXT
0.0005	11.02	20.88	32.96	38.71	57.96
0.001	7.92	14.88	23.40	31.31	43.04
0.002	5.82	10.48	16.59	23.21	32.68
0.005	3.84	6.94	10.72	14.85	22.06
0.01	3.03	5.05	7.52	10.67	15.48
0.02	2.48	3.74	5.50	7.56	11.01
0.05	2.11	2.65	3.57	4.94	6.89

(d) MAPEs for STR, (proportional) stratification by `QTYPE`

Table 4.3: Relative precision for the decomposition of DNS traffic into the 5 most frequent resource records

from a sample become even at low sampling fractions.

Confidence Limits

By again interpreting the MAPE as a measure for variation and using the Central Limit Theorem, confidence bounds on the expected errors can be created by multiplying the MAPE with a z-quantile.
Using this logic on the decomposition considering time, if 100 samples using 1% of the data were drawn using SYS, only in 5 cases would the errors for `AAAA` be larger than 9%. In the case of viewing the whole day, on repeated sampling, the deviation from `AAAA`'s true proportion of traffic would also only be larger than 3.6% for a 1% SYS sample in 5 of 100 cases.

4.2.3 Number of active resolvers per minute

Another metric that is of interest in DNS traffic analysis is the number of resolvers active at a given time. After investigating the questions of how many queries arrived and what was requested at a given time, this feature looks at how many clients were involved in creating the observed traffic pattern.
As was discussed in section 2.4, tracking the number of unique source IP addresses and

TYPE	True	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
A	158952661	1.89	1.89	1.91	1.90	1.87	1.88	1.89
AAAA	41720900	1.77	1.77	1.72	1.68	1.75	1.76	1.73
MX	18876168	1.45	1.55	1.25	1.50	1.55	1.45	1.42
NS	14943446	1.33	1.02	1.02	0.96	0.95	1.03	1.00
TXT	7345880	1.34	0.87	0.84	0.53	0.53	0.55	0.61

(a) MAPEs for SRS

TYPE	True	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
A	158952661	1.92	1.83	1.88	1.88	1.88	1.89	1.88
AAAA	41720900	1.67	1.88	1.69	1.76	1.78	1.75	1.75
MX	18876168	1.29	1.46	1.52	1.37	1.43	1.43	1.45
NS	14943446	1.32	1.04	1.08	1.08	1.05	1.01	1.02
TXT	7345880	1.20	0.97	0.52	0.62	0.52	0.63	0.62

(b) MAPEs for SYS

TYPE	True	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
A	158952661	1.63	1.91	1.91	1.90	1.89	1.90	1.89
AAAA	41720900	1.63	1.84	1.74	1.71	1.76	1.74	1.76
MX	18876168	1.36	1.38	1.51	1.46	1.47	1.44	1.48
NS	14943446	1.23	1.24	1.02	1.01	1.15	1.04	1.06
TXT	7345880	0.89	1.21	0.63	0.70	0.53	0.54	0.58

(c) MAPEs for STR, (proportional) stratification by minute bins

TYPE	True	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
A	158952661	1.88	1.88	1.88	1.88	1.88	1.88	1.88
AAAA	41720900	1.75	1.75	1.75	1.75	1.75	1.75	1.75
MX	18876168	1.46	1.46	1.46	1.46	1.46	1.46	1.46
NS	14943446	1.03	1.01	1.03	1.02	1.02	1.02	1.02
TXT	7345880	0.57	0.57	0.58	0.58	0.57	0.58	0.58

(d) MAPEs for STR, (proportional) stratification by `QTYPE`

Table 4.4: Relative precision for the decomposition of DNS traffic into 5 of the most frequently requested resource records per day

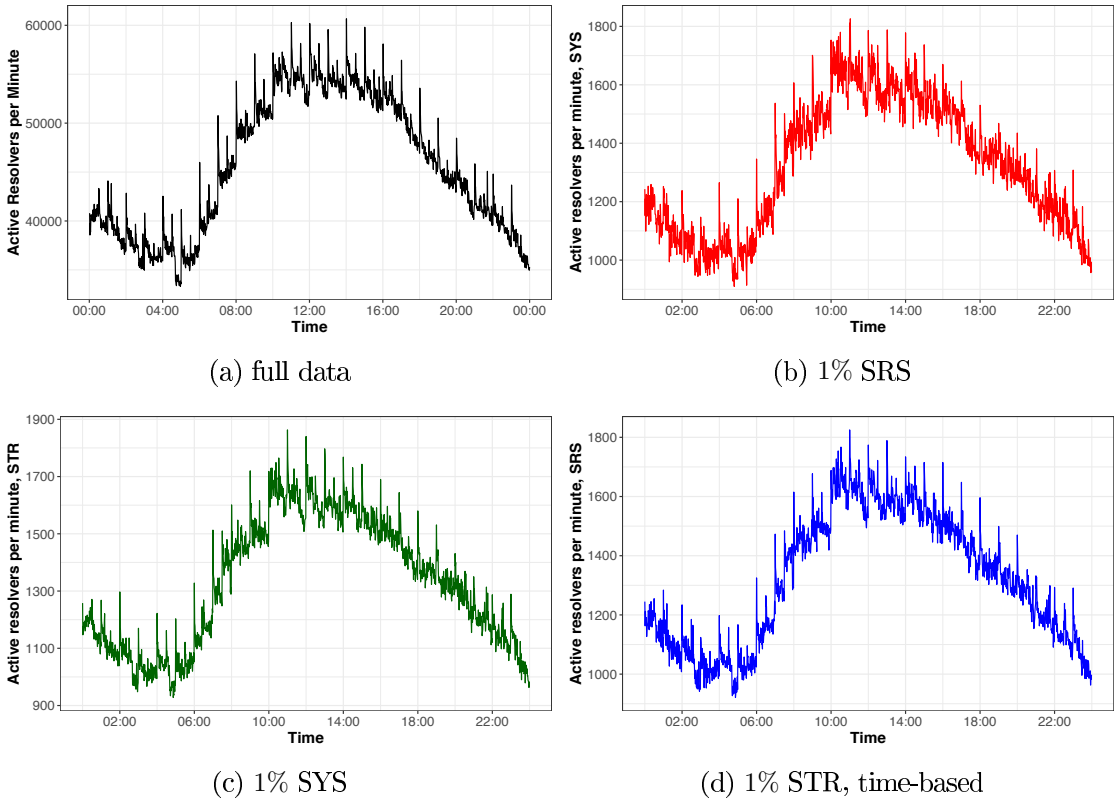
the amount of traffic they generate can be a used to flag malicious domains. In a sampling context, this corresponds to the estimation of the number of distinct *classes* that make up the population. Unfortunately, a reliable way of estimating the (unknown) number

of distinct attributes in a population is a question in statistical research that has yet to be answered conclusively.
Haas et al. [1995] discuss several estimators for distinct value estimation, but their application to the analysis of DNS traffic data, notoriously skewed, bursty and with many classes of sizes less than 10, was ultimately deemed to be beyond the scope of this thesis. Using the approach sketched in chapter 3, where the estimate is scaled up by means of the sampling fraction $\frac{N}{n}$, leads to a severe over-estimation of the number of distinct resolvers active in a given time slot. Figure 4.7 juxtaposes the *unadjusted* results of each sampling method with the actual observed number of active resolvers per minute (4.7a) using the small Vienna-based name server instance for the figures.
The main issue here is that each minute slot would need to receive its own weight for the population estimate. But as the number of resolvers shows a lot of variation and instationarity, as well as a changing number of queries sent, it is evident that there is no single factor that would work for all time slots; calculating individual weights would require knowledge of the true number of active resolvers, the very thing to be estimated.

Figure 4.7a shows that the number of active resolvers also follows the diurnal pattern, although with much less variation and spikes as compared to figure 4.1, although now the increases in traffic occuring at the start of every full and half hour become even more clearly visible.
Although the scale of the sample estimates could not be corrected for lack of a reasonable scaling factor, the contour of the curves, particularly for SYS and STR do follow the overall trend of the true values, albeit with increased variation. So even if the rescaling might prove to be a topic for further research into distinct value sampling, the location of spikes in the number of active resolvers might still be indicative for network traffic analysts.

Total proportion of traffic from most active resolvers

While sampling provides little benefit for the estimation of distinct, rare or unevenly distributed features of a population, when it comes to the calculation of summary aggregates, the prospects become better again: If the focus is switched from calculating an estimate for the number of distinct resolvers for each of the 1440 minutes of a day to the 10 most active resolvers and the amount of traffic they generated over the whole day, sampling can make up lost ground. Unfortunately, even at this level of aggregation, it is very difficult to accurately assess the total number of unique resolvers in traffic.
Still, it is interesting to note, that the daily total of queries is generated by *less than one*



(a) full data

(b) 1% SRS

(c) 1% SYS

(d) 1% STR, time-based

Figure 4.7: Comparison true number of active resolvers per minute to unadjusted sampled estimates (small Vienna-based name server instance)

million unique resolvers, 995.973 to be exact, although more than half sent less than 10 queries.

Figure 4.8 could be indicative of a possible relationship ruled by *Zipf's Law* (Zipf [1950]), where the frequency of queries by a resolver is inversely proportional to the resolver's rank in the frequency table. Further investigation into this relationship might be a starting point in finding a way to get an estimate for the number of distinct resolvers active in DNS traffic. This relationship could also be exploited for the detection of abnormal traffic patterns: A linguistic law related to Zipf's with a similar thrust called *Heaps' Law* was already investigated by Yuchi et al. [2010] and used to detect a malicious pattern in Chinese DNS traffic.

Table 4.5 lists the relative mean precision of 20 repetitions for the estimate of the total amount of traffic generated by the 10 most active resolvers over the whole day. Due to privacy concerns, the resolvers' IP addresses have been censored. As it is likely that the daily trend will affect precision, STR again uses proportional samples from each minute

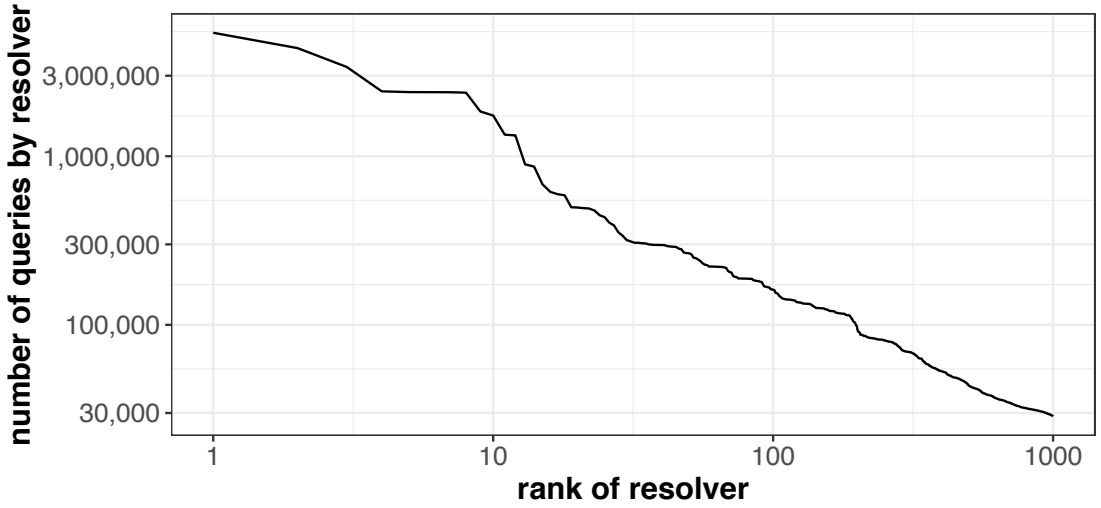


Figure 4.8: Log-Log-Plot: Number of queries vs. the rank of the client, 1000 most active resolvers (base 10)

bin.

At sampling fractions below 1%, there is no clearcut frontrunner, although at higher sampling fractions SYS and STR pull ahead of SRS, supporting the hypothesis of some dependency on time.

Confidence Limits

By multiplying with a z-quantile as discussed in previous sections, approximate confidence bounds for the MAPEs can be established. In the case of repeatedly drawing 100 1% SYS samples and calculating the amount of traffic for the most active resolver in the sample, the estimate's deviation will only be larger than 0.53% in 5 instances.

IP address	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
85.X-X-27	5393198	2.09	2.28	1.62	1.05	0.43	0.21	0.25	0.15
2a01:X-X-1	4375505	1.69	2.18	0.75	1.00	0.57	0.21	0.11	0.24
148.X-X-115	3380767	1.31	0.96	1.14	0.53	0.24	0.56	0.26	0.20
2a01:X-X-2	2423992	0.94	2.29	1.07	0.75	0.82	0.26	0.22	0.15
213.X-X-164	2399556	0.93	2.70	0.59	0.91	0.42	0.39	0.32	0.17
213.X-X-174	2396808	0.93	1.51	1.08	0.90	0.57	0.45	0.35	0.18
213.X-X-173	2394996	0.93	2.03	2.07	0.54	0.78	0.72	0.27	0.19
213.X-X-172	2380430	0.92	2.55	1.83	0.84	0.68	0.45	0.24	0.20
195.X-X-1	1841169	0.71	1.92	1.85	0.84	0.52	0.23	0.51	0.10
176.X-X-71	1742568	0.67	2.47	2.38	0.75	0.69	0.73	0.59	0.31

(a) MAPEs for SRS

IP address	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
85.X-X-27	5393198	2.09	0.30	1.35	0.59	0.43	0.27	0.16	0.13
2a01:X-X-1	4375505	1.69	1.78	1.17	0.56	0.40	0.45	0.23	0.20
148.X-X-115	3380767	1.31	2.25	2.02	0.63	0.50	0.12	0.14	0.07
2a01:X-X-2	2423992	0.94	1.44	1.38	1.50	0.21	0.41	0.16	0.07
213.X-X-164	2399556	0.93	3.55	1.39	1.90	0.53	0.45	0.48	0.18
213.X-X-174	2396808	0.93	1.03	1.84	0.41	0.66	0.64	0.32	0.11
213.X-X-173	2394996	0.93	1.69	1.16	0.96	0.55	0.34	0.17	0.33
213.X-X-172	2380430	0.92	2.33	1.78	1.62	0.61	0.21	0.41	0.20
195.X-X-1	1841169	0.71	2.88	1.59	0.25	0.78	0.39	0.32	0.25
176.X-X-71	1742568	0.67	0.98	1.12	0.89	0.42	0.57	0.40	0.28

(b) MAPEs for SYS

IP address	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
85.X-X-27	5393198	2.09	0.78	0.75	0.79	0.59	0.36	0.16	0.12
2a01:X-X-1	4375505	1.69	1.50	0.70	0.39	0.76	0.27	0.17	0.11
148.X-X-115	3380767	1.31	1.67	0.91	0.74	0.36	0.21	0.33	0.24
2a01:X-X-2	2423992	0.94	2.27	1.54	1.06	0.30	0.38	0.09	0.27
213.X-X-164	2399556	0.93	2.16	1.76	0.72	0.71	0.13	0.46	0.33
213.X-X-174	2396808	0.93	3.36	0.61	1.27	0.60	0.53	0.24	0.09
213.X-X-173	2394996	0.93	4.18	1.91	1.89	0.21	0.79	0.36	0.11
213.X-X-172	2380430	0.92	2.43	0.92	2.18	0.69	0.66	0.22	0.22
195.X-X-1	1841169	0.71	2.85	0.91	1.82	1.28	0.78	0.29	0.26
176.X-X-71	1742568	0.67	2.90	1.60	1.15	1.23	0.70	0.40	0.29

(c) MAPEs for STR, (proportional) stratification by minute bins

Table 4.5: Relative precision for the estimation of the 10 most active resolvers

4.3 Aggregation and Filter-based Features

As was discussed at the beginning of this thesis, sampling can be a useful tool for the long-term storage of DNS traffic data, as it combines aspects of *aggregation* and *filtering*, while maintaining a level of granularity not typically present in both other approaches. Of course, as the sample sizes get smaller, only the broad strokes of traffic characteristics are kept.
In the subsequent sections the previously established way to calculate approximate confidence limits will no longer be addressed directly, but the calculation procedure remains the same as above.

4.3.1 Queries for a specific domain name

A feature relevant to DNS traffic analysis is the amount of queries for a certain domain, i.e. for the post-mortem detection of DDos-attacks or to find possible phishing domains. Naturally, for domains receiving only few hits, any sampling method will be unreliable, but especially for popular sites – or sites suddenly receiving more traffic than usual, drawing and storing a sample can suffice for accurate enough estimates.
For the subsequent example, queries containing the string *google* will be analysed, particularly the domains containing this string that receive the most traffic. Of course, a sample could also satisfyingly be used to search any string combination in the `QNAME` field, provided it occured frequently enough.

In table 4.6 (results again for 20 repetitions) the strengths and downsides of using a sampling method to estimate the daily queries for a specific domain and its subdomains can be seen: The more frequently a domain is requested, the better its true request frequency can be estimated.
Queries for *www.google.at* made up about 0.4% of total traffic in the .at-zone on May 6th 2015, and starting at sampling fractions greater than 0.0005% of the total data, the mean absolute percentage error already falls below 10%, with SYS and time-based STR performing the best. On the other hand, for the other domains, a notable increase in precision requires sample sizes of 1% and beyond, and even then infrequently requested domains often are not in the sample. A possible rule of thumb is, that if the true occurence lies beneath the sampling fraction, a precise estimate will be the exception rather than the rule.
In conclusion, unless something unusual is afoot, the amount of traffic a domain receives should be a random attribute, which is reflected in the relative precisions reported above. Still, should this behaviour change from one moment to the other, SYS and time-based

QNAME	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
www.google.at.	101207	0.039	9.86	8.16	6.22	3.29	2.51	2.12	1.30
blogsearch.google.at.	6275	0.0024	58.69	25.58	19.92	14.38	9.00	5.90	3.31
translate.google.at.	3280	0.0013	67.36	39.26	31.43	16.93	12.38	7.91	5.56
maps.google.at.	3118	0.0012	74.02	50.85	33.98	19.94	15.84	10.70	6.56
accounts.google.at.	2969	0.0011	78.63	56.65	36.97	22.30	17.08	11.79	7.82
id.google.at.	2071	0.0008	93.14	88.11	61.39	43.83	32.86	17.69	13.71
books.google.at.	1627	0.0006	22.93	84.39	145.85	38.54	1.66	4.49	3.26
news.google.at.	1124	0.0004	96.76	80.63	44.45	36.01	17.72	17.35	11.57
accounts.google.com.sms.at.	1077	0.0004	100.00	80.88	80.66	40.99	25.53	19.95	14.07
clients1.google.at.	910	0.0003	116.92	100.00	100.00	100.00	100.00	100.00	100.00

(a) MAPEs for SRS

QNAME	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
www.google.at.	101207	0.039	10.81	7.94	5.49	3.75	2.36	1.62	1.05
blogsearch.google.at.	6275	0.0024	41.00	31.16	23.45	15.24	9.92	6.10	3.94
translate.google.at.	3280	0.0013	55.73	40.63	27.53	17.93	13.24	7.10	5.21
maps.google.at.	3118	0.0012	76.05	54.49	34.11	24.45	15.18	9.90	6.50
accounts.google.at.	2969	0.0011	78.97	55.73	35.69	25.07	15.98	10.92	7.80
id.google.at.	2071	0.0008	93.14	44.86	3.43	15.89	1.40	3.43	12.02
books.google.at.	1627	0.0006	99.42	61.51	41.16	26.07	17.38	16.23	8.65
news.google.at.	1124	0.0004	100.00	75.59	42.29	28.06	22.01	16.54	8.79
accounts.google.com.sms.at.	1077	0.0004	115.93	91.87	65.71	37.36	25.93	16.64	11.09
clients1.google.at.	910	0.0003	135.66	100.00	100.00	100.00	100.00	100.00	100.00

(b) MAPEs for SYS

QNAME	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
www.google.at.	101207	0.039	8.88	6.43	6.63	5.32	2.87	1.90	1.13
blogsearch.google.at.	6275	0.0024	47.53	35.14	18.25	13.43	9.32	6.97	4.74
translate.google.at.	3280	0.0013	62.68	42.00	32.07	14.83	11.83	7.05	4.94
maps.google.at.	3118	0.0012	63.68	44.49	32.41	17.44	12.41	8.12	6.74
accounts.google.at.	2969	0.0011	71.31	49.71	32.77	18.54	14.10	8.36	6.95
id.google.at.	2071	0.0008	73.11	49.82	34.11	19.74	15.08	12.47	7.55
books.google.at.	1627	0.0006	100.00	62.19	46.73	25.80	18.63	13.52	7.57
news.google.at.	1124	0.0004	106.76	81.14	46.90	25.86	20.23	14.26	9.95
accounts.google.com.sms.at.	1077	0.0004	109.90	86.37	61.21	42.75	34.29	19.07	11.87
clients1.google.at.	910	0.0003	117.91	100.00	100.00	100.00	100.00	100.00	100.00

(c) MAPEs for STR, (proportional) stratification by minute bins

Table 4.6: Relative precision for the most frequently requested *google*-domains

STR would be particularly well-suited to capture these shifts.

4.3.2 Query Counts by Source Country

Another way to decompose traffic is by the country from where the query originated. This allows insight in the makeup of the clients accessing the .at-zone, but can also be used to identify irregular or malicious traffic patterns, as sudden spikes in queries for a certain domain from a country or region that usually generates less traffic might be indicative of a DDos-attack or a phishing scheme.

Figure 4.9 visualizes the countries of query origin and their respective proportion of traffic, with the U.S., Germany, Austria, the Netherlands and China being responsible for the bulk of traffic hitting nic.at's name servers. Table 4.7 shows the relative precisions

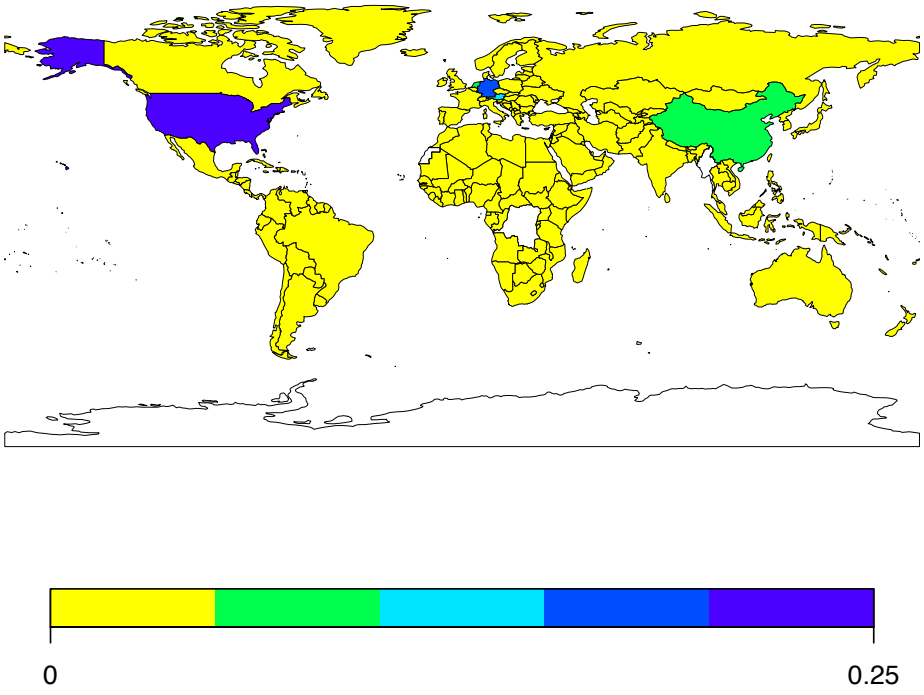


Figure 4.9: Heatmap showing source countries for queries

for each method in estimating the proportion of traffic originating from the 10 most active countries. Stratification was again conducted with regards to time of arrival, with a second variant drawing proportional samples directly from each source country group. It is therefore not surprising, that the latter type of stratification results in estimation errors of practically zero, since stratifying by the very characteristic to be estimated reduces within-sample variation enormously, even at low sampling fractions.
By looking at the results for time-based STR and SYS, which both also perform quite well, it appears there is no particular time-dependent trend for countries. Thus, it is likely that – unless an abnormal traffic pattern occurs – the geographical origin of a query is essentially a randomly ordered attribute, which as was discussed in chapter 3 makes the different sampling methods practically indistinguishable. This is also reflected in the results for SRS. However, if a suitable stratification variable is used, STR outperforms all other approaches, as per theorem 3.5.2, which as already mentioned is reflected in table 4.7.

Country	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
United States	60240553	23.32	0.34	0.20	0.15	0.04	0.06	0.03	0.01
Germany	44318337	17.16	0.64	0.83	0.20	0.09	0.10	0.11	0.07
Austria	32849955	12.712	0.97	0.57	0.14	0.44	0.10	0.10	0.06
Netherlands	16602190	6.43	0.67	1.16	0.13	0.21	0.27	0.25	0.04
China	13726583	5.31	1.77	0.35	0.47	0.17	0.26	0.19	0.15
Russia	9633069	3.73	1.26	1.30	0.45	0.20	0.22	0.15	0.13
Belgium	5672110	2.19	0.64	0.81	1.08	0.23	0.40	0.11	0.21
Poland	5471359	2.12	0.73	1.81	0.68	0.55	0.32	0.08	0.19
France	5103135	1.98	3.59	0.61	0.78	0.82	0.36	0.40	0.07
United Kingdom	4559826	1.77	1.95	0.33	0.69	0.74	0.31	0.30	0.16

(a) MAPEs for SRS

Country	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
United States	60240553	23.32	0.18	0.14	0.17	0.10	0.06	0.09	0.05
Germany	44318337	17.16	0.14	0.46	0.19	0.10	0.06	0.09	0.01
Austria	32849955	12.72	0.51	0.50	0.09	0.24	0.12	0.06	0.02
Netherlands	16602190	6.43	0.82	0.20	0.34	0.07	0.45	0.12	0.15
China	13726583	5.31	0.64	0.35	1.42	0.47	0.29	0.12	0.01
Russia	9633069	3.73	0.99	0.46	0.85	0.57	0.18	0.18	0.07
Belgium	5672110	2.19	1.75	0.88	0.54	0.52	0.45	0.23	0.07
Poland	5471359	2.11812	0.33	1.38	0.93	0.50	0.12	0.05	0.13
France	5103135	1.97	1.23	1.14	1.90	0.39	0.32	0.60	0.08
United Kingdom	4559826	1.77	0.90	0.83	0.35	0.74	0.61	0.21	0.21

(b) MAPEs for SYS

Country	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
United States	60240553	23.32	0.39	0.16	0.21	0.08	0.05	0.05	0.04
Germany	44318337	17.16	0.27	0.28	0.43	0.13	0.09	0.08	0.03
Austria	32849955	12.72	0.68	0.20	0.48	0.20	0.11	0.04	0.01
Netherlands	16602190	6.43	0.45	0.33	0.25	0.36	0.01	0.13	0.05
China	13726583	5.31	0.75	0.86	0.25	0.48	0.32	0.15	0.08
Russia	9633069	3.73	1.30	0.72	0.68	0.31	0.33	0.11	0.09
Belgium	5672110	2.19	3.28	1.26	0.79	0.10	0.40	0.06	0.06
Poland	5471359	2.12	2.42	0.56	0.34	0.25	0.16	0.03	0.09
France	5103135	1.97	2.56	0.76	0.60	0.16	0.12	0.22	0.06
United Kingdom	4559826	1.77	2.57	0.63	1.24	0.67	0.22	0.18	0.30

(c) MAPEs for STR, (proportional) stratification by minute bins

Country	True Traffic	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
United States	60240553	23.32	0.00	0.00	0.00	0.00	0	0	0
Germany	44318337	17.26	0.01	0.00	0.00	0.00	0	0	0
Austria	32849955	12.82	0.01	0.00	0.00	0.00	0	0	0
Netherlands	16602190	6.43	0.01	0.01	0.00	0.00	0	0	0
China	13726583	5.31	0.01	0.00	0.00	0.00	0	0	0
Russia	9633069	3.73	0.01	0.01	0.00	0.00	0	0	0
Belgium	5672110	2.19	0.03	0.02	0.01	0.00	0	0	0
Poland	5471359	2.12	0.01	0.01	0.00	0.00	0	0	0
France	5103135	1.98	0.06	0.00	0.01	0.01	0	0	0
United Kingdom	4559826	1.77	0.00	0.02	0.01	0.00	0	0	0

(d) MAPEs for STR, (proportional) stratification by source country

Table 4.7: Relative precision for the decomposition of DNS traffic into the 10 most frequent countries of query origin

4.3.3 Adoption of IPv6

In 1998, Deering and Hinden [1998] from the Internet Engineering Task Force (IETF) introduced the Internet Protocol Version 6 (IPv6), to solve IPv4's looming problem of address exhaustion although in 2010, Castro et al. [2010] still commented on the slow deployment of IPv6 support, in particular for the 13 root servers.
This section now investigates what proportion of arriving queries uses which protocol version and whether this information can accurately be estimated using sampling methods.
Table 4.8 summarises the results.

IP version	True Amount	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
IPv4	233271468	0.90	1.89	1.90	1.91	1.91	1.90	1.91	1.91
IPv6	29559688	0.11	0.57	0.55	0.29	0.31	0.26	0.23	0.20

(a) MAPEs for SRS

IP version	True Amount	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
IPv4	233271468	0.90	1.94	1.93	1.92	1.91	1.92	1.91	1.91
IPv6	29559688	0.11	0.83	0.36	0.35	0.27	0.16	0.22	0.23

(b) MAPEs for SYS

IP version	True Amount	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
IPv4	233271468	0.90	2.01	1.98	1.91	1.91	1.92	1.92	1.92
IPv6	29559688	0.11	0.66	0.41	0.30	0.31	0.25	0.26	0.23

(c) MAPEs for STR, (proportional) stratification by minute bins

IP version	True Amount	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
IPv4	233271468	0.90	1.91	1.91	1.91	1.91	1.91	1.91	1.91
IPv6	29559688	0.11	0.22	0.24	0.23	0.23	0.23	0.23	0.23

(d) MAPEs for STR, (proportional) stratification by IP version

Table 4.8: Relative precision for the proportion of IPv4 and IPv6 usage

The tables seen in 4.8 display the total number of queries using the respective IP protocol version and their respective proportion of total traffic.
The results for increasing sampling sizes list the mean absolute percentage errors from the reported *totals* for 20 drawings for each method-sampling fraction combination.
What can be learned from the tables is that the the IP version is again likely a randomly ordered attribute that does not depend on any time-based patterns, as all three methods again show no marked differences. Interestingly, drawing a proportional sample of the subpopulations using either IPv4 or IPv6 did not increase the estimate's precision.

4.3.4 Distribution of TTLs

Another significant property of a DNS query is its *time-to-live* (TTL), the time for how long it is valid. It is possible to match the TTL of a query to different operating systems by their usual default settings.
The default TTL-setting for Linux- or Unix-based operating systems for example is set to 64 seconds, while Windows systems have a default of 128 seconds[7]. Thus, by looking at the distribution of TTLs, it is possible to draw inference on the usage of certain operating systems. Furthermore having accurate statistics on the distribution of the TTLs can be a useful tool, as it was found that some malicious domains can be identified by their TTL settings.
A sampling method that accurately reflects this, would be a useful tool for quick exploratory analysis in a big dataset as well as it could reduce storage requirements. Before analysis, the TTLs resulting from sampling are grouped into slices of length 32, with the resulting counts then being corrected for sampling.

Table 4.9 shows the resulting relative precisions for the different sampling methods, repeated 20 times. It is apparent that Linux/Unix-based operating systems dominate traffic, which seems reasonable, as for example the majority of smartphones have operating systems based on Android, a Linux derivative.
Also, the operating systems of choice for large-scale Internet operations are usually Linux- or Unix-based, not least because they are cheaper than Windows and are less resource-intensive.
The pattern that was already observable with queries for Google-domains repeats here: The more frequent a characteristic is present in the data, the better it is retained through sampling. Also, there is again no clearcut frontrunner which indicates that TTL is a randomly ordered attribute.
For queries with a TTL close to 64, at sampling fractions of only 0.0005% of the full data the estimate's deviation from the true characteristic is already practically zero, regardless of method, while for queries with TTLs of 160 or higher most reported sampling fractions are actually higher than their actual occurence in the full dataset.

[7] See `https://subinsb.com/default-device-ttl-values/` for an extensive list.

TTL	True Amount	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
32	124364	0.05	6.62	7.30	4.79	3.17	2.66	1.23	1.10
64	229120360	88.70	0.06	0.06	0.04	0.02	0.02	0.01	0.01
96	2443440	0.95	2.05	1.70	1.21	0.73	0.51	0.39	0.29
128	13266229	5.14	0.81	0.72	0.43	0.31	0.27	0.17	0.09
160	2409	0.00	67.36	51.89	35.94	23.25	16.98	9.96	6.10
192	3824	0.00	61.84	37.07	24.31	14.76	12.43	7.98	5.56
224	41981	0.02	24.05	10.96	8.69	5.77	4.46	2.66	2.51
256	13312614	5.15	0.79	0.70	0.50	0.25	0.13	0.13	0.11

(a) MAPEs for SRS

TTL	True Amount	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
32	124364	0.05	8.31	6.02	4.03	3.89	2.65	1.78	0.84
64	229120360	88.70	0.11	0.06	0.04	0.02	0.02	0.01	0.01
96	2443440	0.95	2.19	2.09	1.08	0.82	0.54	0.42	0.21
128	13266229	5.14	1.21	0.64	0.47	0.32	0.19	0.17	0.12
160	2409	0.00	66.60	38.30	33.21	28.23	20.51	12.36	7.97
192	3824	0.00	56.30	43.68	37.46	23.87	13.52	8.10	4.13
224	41981	0.02	17.38	10.24	7.25	4.92	3.74	2.35	1.32
256	13312614	5.15	0.86	0.70	0.54	0.29	0.13	0.13	0.06

(b) MAPEs for SYS

TTL	True Amount	% of Total	0.0005	0.001	0.002	0.005	0.01	0.02	0.05
32	124364	0.05	7.54	8.17	5.42	2.88	2.29	1.46	1.04
64	229120360	88.70	0.22	0.07	0.05	0.03	0.02	0.02	0.02
96	2443440	0.95	2.33	0.96	1.10	0.78	0.57	0.30	0.22
128	13266229	5.14	1.06	0.58	0.52	0.25	0.19	0.15	0.10
160	2409	0.00	78.12	57.74	30.76	21.66	12.15	9.58	7.88
192	3824	0.00	62.76	27.07	26.27	19.86	14.77	10.20	5.63
224	41981	0.02	21.91	11.08	8.64	4.64	3.63	2.34	2.00
256	13312614	5.15	0.73	0.64	0.48	0.21	0.20	0.13	0.10

(c) MAPEs for STR, (proportional) stratification by minute bins

Table 4.9: Relative precision for distribution of TTLs

5 Conclusion

The stated goals of this thesis were to a) investigate whether estimates from probability sampling methods provide accurate enough results for DNS traffic analysis tasks as conducted by registry operators such as nic.at and b) use the test features to provide a glimpse into the characteristics of DNS traffic in Austria.

DNS traffic generates large amounts of data, and with the projected increase in devices connected to the Internet, this growth will continue further - as will the need for network analysis, maintenance and defense against malicious elements.

Up to now, to the best of the author's knowledge, research into DNS traffic analysis put little focus on the applicability of sampling methods and more on Big Data solutions, which when combined though might lead to even greater gains in computation speeds for a comparatively little price in accuracy than each tool on its own.

As was shown in this thesis, a sample can preserve many important characteristics of DNS traffic with the degree of precision only dependent on sample size, while retaining arguably more information than any aggregated or filtered dataset.

Concerning long-term storage, if the sample is for example converted to a data format better suited for storage than pcap and then stored by a datastreaming warehouse such as ENTRADA, there is also great potential for added value from sampling.

In order to assess the feasibility of probability sampling, after introducing the *Domain Name System* (DNS), related research from general network traffic analysis and reviewing relevant mathematical theory, several features used in network traffic analysis were analysed using the three most emblematic statistical sampling methods, namely *simple random sampling* (SRS), *systematic sampling* (SYS) and *stratified sampling* (STR).

One of the key findings from theory was the importance of whether the characteristic to be estimated can be considered to be randomly ordered or not. If stratification by the feature or something closely related to it did not improve the precision of the estimates, according to theorem 3.5.3 the feature was deemed to be essentially random, which was reflected in SRS, SYS and STR resulting in practically equivalent precisions.

This was found for attributes such as the type of the resource record (RR) requested, or the used IP version, as drawing a STR sample that took into account their overall

proportion of total traffic led to no discernable improvement in relative precision. One exception was for the most frequent countries of query origin, although even there time-based stratification, SRS and SYS led to results with on average less than 1% error for the 10 most frequent countries for sample sizes starting at a fraction of a percent of the total.
On the other hand, particularly for estimating properties where the order in which queries arrived was of importance, it was found that the self-similar nature of DNS traffic, which expresses itself in high autocorrelation, leads to large gains in precision for SYS and time-based STR, with SYS outperforming its competitors by a large margin.
Using only 0.002% of a dataset with more than 250 million rows ($n \approx 500.000$), systematic sampling could repeatedly estimate the per minute query arrival with a mean absolute error (MAPE) of less than 1%. Even with approximate 97.5% confidence bounds, the MAPE remained close to 1%.
In summary, the sampling method that consistently performed well and required the least computational overhead was found to be *systematic sampling*, as it is by design best suited for taking the temporal structure of the query arrival process into account and otherwise performs competitively to the other two methods. There are instances where stratified sampling leads to better results, but at the cost of more complex queries into the data. Furthermore, SYS has the advantage that it could also easily be deployed in realtime at name server level.

5.1 Limitations

The analysis presented in chapter 4 also highlighted the unsurprising downsides of using sampling methods: For randomly ordered attributes, where there is no reasonable stratification variable or underlying systematic pattern, characteristics that occur only rarely are likely underrepresented, or overrepresented, depending on how many observations were randomly selected into the sample.
This was particularly visible in section 4.3.1, where some of the smaller, less frequented Google subdomains were either not represented in the estimate at all (MAPE 100%) or had a high related MAPE.
Another point where sampling can be only of minor use, is in the estimation of the number of unique resolvers active in traffic at a given time as it would require knowledge from the very thing to be estimated to accurately rescale the estimates to population level. However, it was found that the overall trend of the unadjusted estimates might still be

useful in assessing irregular behaviour.

5.2 Characteristics of the .at-Zone

Concerning characteristics of Austrian DNS traffic, the query arrival process and several ways to decompose it into components for further analysis were presented. In the dataset made available by nic.at, some peculiar traffic patterns resulting from spikes in `TXT` queries were discussed, and it was shown that these patterns were retained even at low sampling fractions.
Also, it was shown that although nic.at's name servers registered over 250 million queries arriving, the number of unique resolvers responsible for them was found to be below 1 million, with indications of a relationship possibly described by Zipf's law.
The bulk of the queries arriving at nic.at's name server originated from the U.S., Austria, its neighbour Germany, China and the Netherlands. Furthermore, it was found that for this dataset approximately 11% of all queries used IPv6.
These features were also well estimated by the presented methods.

5.3 Future Work

DNS traffic analysis is a dynamic field and allows for a myriad of potential research. Due to limitations in time and scope, the author's explorations into the relationship between the query arrival process and their interarrival times had to be cut short, but their investigation might yield new insights to identify irregular traffic by the time passing between subsequent packets. Linking interarrival times and query arrival by means of a stochastic process could also be used to create a parametric model of DNS traffic, which again could be used for detecting irregularities, but also for network maintenance and resource planning.
Better understanding the query arrival process and its composition could also allow statistical methods for outlier detection, forecasting or predictive maintenance to be applied to DNS traffic data.
Another possible venue for research could be extending research on distinct value sampling or species diversity to DNS traffic to better estimate the number of unique resolvers active in a certain period of time, a feature that is vital to assess for example *source port randomization* or again to identify malicious domains that use many different IP addresses for the same domain.

5.4 Epilogue

Sampling can be an invaluable addition to the toolbelt of network traffic engineers, especially when the task involves finding the approximately quite right amount of needles in a haystack fast – given that there are at least a minimum number of needles in the haystack – or whether the haystack contains notably more needles than usual.
If the task involves finding out whether the needles in the haystack can be grouped into knitting needles, sewing needles and the odd pin, or if the *one* needle in the stack is red – then it might be best to also have a Big Data engine at hand.

Bibliography

P. Albitz and C. Liu. *DNS and BIND*. Help for System Administrators. O'Reilly, 1997. ISBN 9781565922365.

P.D. Amer and L.N. Cassel. Management of sampled real-time network measurements. In *[1989] Proceedings. 14th Conference on Local Computer Networks*. IEEE Comput. Soc. Press, 1989. doi:10.1109/lcn.1989.65244.

M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a Dynamic Reputation System for DNS. pages 273–290, 2010. doi:10.1.1.172.8010.

J. Beran, Y. Feng, S. Ghosh, and R. Kulik. *Long-Memory Processes*, pages 529–554. 2013. ISBN 978-3-642-35511-0. doi:10.1007/978-3-642-35512-7_6.

L. Bilge, E. Kirda, C. Kruegel, M. Balduzzi, and S. Antipolis. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *In Annual Network and Distributed System Security Symposium (NDSS)*, 2011. doi:10.1.1.185.474.

P. Billingsley. *Probability and Measure*. Wiley Series in Probability and Statistics. Wiley, 2012. ISBN 9781118341919.

S. Castro, D. Wessels, M. Fomenkov, and K. C. Claffy. A Day at the Root of the Internet. *ACM SIGCOMM Computer Communication Review*, 38(5):41–46, 2008. doi:10.1145/1452335.1452341.

S. Castro, M. Zhang, W. John, D. Wessels, and K. C. Claffy. Understanding and Preparing for DNS Evolution. In *Traffic Monitoring and Analysis*, pages 1–16. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-12365-8_1.

B. Choi and Z. Zhang. Adaptive Random Sampling for Traffic Volume Measurement. *Telecommunication Systems*, 34(1-2):71–80, 2006. doi:10.1007/s11235-006-9023-z.

B. Choi, J. Park, and Z. Zhang. Adaptive Random Sampling for Traffic Load Measurement. In *IEEE International Conference on Communications, 2003. ICC '03.* IEEE. doi:10.1109/icc.2003.1203863.

B. Choi, J. Park, and Z. Zhang. Adaptive Random Sampling for Load Change Detection. In *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems - SIGMETRICS '02*. ACM Press, 2002a. doi:10.1145/511334.511376.

B. Choi, J. Park, and Z. Zhang. Adaptive packet sampling for flow volume measurement. *ACM SIGCOMM Computer Communication Review*, 32(3):9–9, 2002b. doi:10.1145/571697.571698.

Cisco Systems, Inc. Cisco Annual Internet Report (2018-2023). Technical report, 2018. URL `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf`.

K. C. Claffy, G. C. Polyzos, and H. Braun. Application of Sampling Methodologies to Network Traffic Characterization. In *Conference proceedings on Communications architectures, protocols and applications - SIGCOMM '93*. ACM Press, 1993. doi:10.1145/166237.166256.

W. G. Cochran. Relative Accuracy of Systematic and Stratified Random Samples for a Certain Class of Populations. *The Annals of Mathematical Statistics*, 17(2):164–177, 1946. doi:10.1214/aoms/1177730978.

W. G. Cochran. *Sampling Techniques, 3rd Edition*. John Wiley, 1977. ISBN 0-471-16240-X.

S. E. Deering and R. M. Hinden. Internet Protocol, Version 6 (IPv6) Specification, Draft. 1998. doi:10.17487/RFC2460.

S. E. Deering and R. M. Hinden. Internet Protocol, Version 6 (IPv6) Specification, Internet Standard. 2017. doi:10.17487/RFC8200.

R. Dodopoulos. DNS-based Detection of Malicious Activity. Master's thesis, 2015.

A. Dogman, R. Saatchi, and S. Al-Khayatt. An Adaptive Statistical Sampling Technique for Computer Network Traffic. In *2010 7th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP 2010)*. IEEE, 2010. doi:10.1109/csndsp16145.2010.5580380.

J. Drobisz and K.J. Christensen. Adaptive Sampling Methods to determine Network Traffic Statistics including the Hurst Parameter. In *Proceedings 23rd Annual Conference on Local Computer Networks. LCN'98 (Cat. No.98TB100260)*. IEEE Comput. Soc. doi:10.1109/lcn.1998.727664.

N. Duffield. Sampling for Passive Internet Measurement: A Review. *Statistical Science*, 19(3):472–498, 2004. doi:10.1214/088342304000000206.

N. Duffield, C. Lund, and M. Thorup. Properties and prediction of flow statistics from sampled packet streams. In *Proceedings of the second ACM SIGCOMM Workshop on Internet measurment - IMW '02*. ACM Press, 2002. doi:10.1145/637201.637225.

N. Duffield, C. Lund, and M. Thorup. Learn more, sample less: Control of volume and variance in network measurement. *IEEE Transactions on Information Theory*, 51(5): 1756–1775, 2005a. doi:10.1109/tit.2005.846400.

N. Duffield, C. Lund, and M. Thorup. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Transactions on Networking*, 13(5):933–946, 2005b. doi:10.1109/tnet.2005.852874.

I. Fette, N. Sadeh, and A. Tomasic. Learning to Detect Phishing Emails. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*. ACM Press, 2007. doi:10.1145/1242572.1242660.

Stratification Strategies for Sampling-based Non-intrusive Measurements of One-way Delay, 2003. FOKUS - Fraunhofer Institute for Open Communication Systems. doi:10.1.1.12.8180.

P. Foremski, O. Gasser, and G. C. M. Moura. DNS Observatory. In *Proceedings of the Internet Measurement Conference*. ACM, 2019. doi:10.1145/3355369.3355566.

P. Haas, J. Naughton, S. Seshadri, and L. Stokes. Sampling-Based Estimation of the Number of Distinct Values of an Attribute. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB 95, page 311322. Morgan Kaufmann Publishers Inc., 1995. ISBN 1558603794. doi:10.5555/645921.673295.

R. J. Hyndman and A. B. Koehler. Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006. doi:10.1016/j.ijforecast.2006.03.001.

J. Jedwab, P. Phaal, and B. Pinna. Traffic Estimation for the largest Sources on a Network, using Packet Sampling with Limited Storage. Technical report, 1992.

T. Karagiannis, Mart Molle, and M. Faloutsos. Long-range dependence - ten years of internet traffic modeling. *Internet Computing, IEEE*, 8:57– 64, 2004. doi:10.1109/MIC.2004.46.

P. R. Krishnaiah and C. R. Rao. *Handbook of Statistics 6: Sampling*. North-Holland, 1988. ISBN 978-0-444-70289-0.

E.L. Lehmann. *Elements of Large-Sample Theory*. Springer Texts in Statistics. Springer New York, 2004. ISBN 9780387985954.

W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic. *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994. doi:10.1109/90.282603.

P. S. Levy and S. Lemeshow. *Sampling of Populations: Methods and Applications*. John Wiley, 2009. doi:10.1002/9780470374597.

A. Liska, G. Stowe, and T. Gallo. *DNS Security: Defending the Domain Name System*. 2016. ISBN 978-0128033067.

Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. C. Claffy. Two days in the life of the DNS anycast root servers. In *Lecture Notes in Computer Science*, pages 125–134. Springer Berlin Heidelberg. doi:10.1007/978-3-540-71617-4_13.

W. G. Madow and L. H. Madow. On the Theory of Systematic Sampling, I. *Ann. Math. Statist.*, 15(1):1–24, 1944. doi:10.1214/aoms/1177731312.

D. Mahjoub, T. Reuille, and A. Toonk. Catching Malware en Masse: DNS and IP Style. *Black Hat USA 2014 Proceedings*, 2014.

P. V. Mockapetris. Domain names: Concepts and facilities. 1983a. doi:10.17487/RFC0882.

P.V. Mockapetris. Domain names: Implementation specification. 1983b. doi:10.17487/RFC0883.

P.V. Mockapetris. Domain names - implementation and specification. 1987. doi:10.17487/RFC1035.

M. Molina, F. Raspall, S. Niccolini, N. Duffield, and T. Zseby. Sampling and Filtering Techniques for IP Packet Selection. Request for Comments, 2009. doi:10.17487/RFC5475.

R. Perdisci, I. Corona, D. Dagon, and W. Lee. Detecting malicious flux service networks through passive analysis of recursive DNS traces. In *2009 Annual Computer Security Applications Conference*. IEEE, 2009. doi:10.1109/acsac.2009.36.

R. Perdisci, I. Corona, and G. Giacinto. Early detection of malicious flux networks via large-scale passive DNS traffic analysis. *IEEE Transactions on Dependable and Secure Computing*, 2012. doi:10.1109/tdsc.2012.35.

J. Quittek, T. Zseby, B. Claise, and S. Zander. Requirements for IP Flow Information Export (IPFIX). 2004. doi:10.17487/RFC3917.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL `https://www.R-project.org/`.

J. M. C. Silva, P. Carvalho, and S. Rito Lima. A Multiadaptive Sampling Technique for Cost-effective Network Measurements. *Computer Networks*, 57(17):3357–3369, 2013. doi:10.1016/j.comnet.2013.07.023.

J. M. C. Silva, P. Carvalho, and S. Rito Lima. A Modular Traffic Sampling Architecture: Bringing Versatility and Efficiency to Massive Traffic Analysis. *Journal of Network and Systems Management*, 25(3):643–668, 2017. doi:10.1007/s10922-017-9404-5.

E. Stalmans, S. O. Hunter, and B. Irwin. Geo-spatial Autocorrelation as a Metric for the Detection of Fast-Flux Botnet Domains. In *2012 Information Security for South Africa*. IEEE, 2012. doi:10.1109/issa.2012.6320433.

United States Computer Emergency Response Team. US-CERT: Vulnerability Note VU800113: Multiple DNS implementations vulnerable to cache poisoning., 2008. URL `https://www.kb.cert.org/vuls/id/800113/`.

R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE Journal on Selected Areas in Communications*, 34(6):1877–1888, 2016. doi:10.1109/jsac.2016.2558918.

R. Villamarin-Salomon and J. C. Brustoloni. Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic. In *2008 5th IEEE Consumer Communications and Networking Conference*. IEEE, 2008. doi:10.1109/ccnc08.2007.112.

M. Wong and W. Schlitt. Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. 2006. doi:10.17487/RFC4408.

M. Wullink, G. C. M. Moura, M.Muller, and C. Hesselman. ENTRADA: A high-performance Network Traffic Data Streaming Warehouse. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016a. doi:10.1109/noms.2016.7502925.

M. Wullink, M. Muller, M. Davids, G. C. M. Moura, and C. Hesselman. ENTRADA: enabling DNS big data applications. In *2016 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2016b. doi:10.1109/ecrime.2016.7487939.

S. Yadav, A. Kumar Krishna Reddy, A.L. Narasimha Reddy, and S. Ranjan. Detecting algorithmically generated malicious domain names. In *Proceedings of the 10th annual conference on Internet measurement - IMC '10*. ACM Press, 2010. doi:10.1145/1879141.1879148.

F. Yarochkin, V. Kropotov, Y. Huang, G. Ni, . Kuo, and I. Chen. Investigating DNS traffic anomalies for malicious activities. In *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*. IEEE, 2013. doi:10.1109/dsnw.2013.6615506.

X. Yuchi, X. Wang, X. Lee, and B. Yan. *A New Statistical Approach to DNS Traffic Anomaly Detection*, pages 302–313. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-17313-4_30.

B. Zdrnja. Security Monitoring of DNS Traffic. CAIDA, 2006. doi:10.1.1.511.7926.

Y. Zhang, J. I. Hong, and L. F. Cranor. CANTINA: A content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*. ACM Press, 2007. doi:10.1145/1242572.1242659.

W. Zheng and T. S. Shyong. Anomaly detection of domain name system (DNS) query traffic at top level domain servers. *Scientific Research and Essays*, 6(18):3858–3872, 2011. doi:10.5897/sre11.439.

G. K. Zipf. Human behavior and the principle of least effort: An introduction to human ecology. *Social Forces*, 28(3):340–341, 1950. ISSN 0037-7732. doi:10.2307/2572028.