A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements

Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras

***Abstract*—The Domain Name System (DNS) is a core component of the Internet. It performs the vital task of mapping human readable names into machine readable data (such as IP addresses, which hosts handle e-mail, etc.). The content of the DNS reveals a lot about the technical operations of a domain. Thus, studying the state of large parts of the DNS over time reveals valuable information about the evolution of the Internet.**

We collect a unique long-term dataset with daily DNS measurements for all domains under the main top-level domains on the Internet (including `.com`, `.net` and `.org`, comprising 50% of the global DNS name space). This paper discusses the challenges of performing such a large-scale active measurement. These challenges include scaling the daily measurement to collect data for the largest TLD (`.com`, with 123M names) and ensuring that a measurement of this scale does not impose an unacceptable burden on the global DNS infrastructure. The paper discusses the design choices we have made to meet these challenges and documents the design of the measurement system we implemented based on these choices. Two case studies related to cloud e-mail services illustrate the value of measuring the DNS at this scale. The data this system collects is valuable to the network research community. Therefore, we end the paper by discussing how we make the data accessible to other researchers.

***Index Terms*—DNS; active measurements; cloud; Internet evolution**

I. Introduction

THE Domain Name System (DNS), plays a crucial role in the day-to-day operation of the Internet. It performs the vital task of translating human readable names – such as `www.example.com` – into machine readable information. Almost all networked services depend on the DNS to store information about the service. Often this information is about what IP address to contact, but also whether or not e-mail received from another host is legitimate or should be treated as spam. Thus, measuring the DNS provides a wealth of data about the Internet, ranging from operational practices, to the stability of the infrastructure, to security. Consider, for example, e-mail handling. In the DNS, the `MX` record type specifies which hosts handle e-mail for a domain. Thus, examining which `MX` records are present can tell us, for example, if e-mail handling for that domain is outsourced to a cloud provider such as Google, Microsoft or Yahoo. Another example is the monitoring of protocol adoption such as IPv6 and DNSSEC. The analysis of `AAAA` or `DNSKEY` resource records can provide ground truth about the adoption of, and operational practices for these protocols over time. Finally, DNS data can also play a vital role in security research, for instance for studying botnets, phishing and malware.

R. van Rijswijk-Deij, M. Jonker, A. Sperotto and A. Pras are with the Design and Analysis of Communications (DACS) group at the faculty for Electrical Engineering, Mathematics and Computer Science of the University of Twente, Enschede, the Netherlands

R. van Rijswijk-Deij is also with SURFnet bv, the National Research and Education Network in Utrecht, the Netherlands

Manuscript received September 9, 2015; revised March 3, 2016.

The DNS has been the focus of, or used in, past measurement studies. These studies, however, had a limited scope, in time, coverage of DNS records or number of domains measured. It remains highly challenging to measure the DNS in a comprehensive, large-scale, and long-term manner. Nonetheless, because this type of measurement can provide such valuable information about the evolution of the Internet, we challenged ourselves to do precisely this. Our research goal is to perform daily active measurements of all domains in the main top-level domains (TLDs) on the Internet (including `.com`, `.net` and `.org`, together comprising 50% of the global DNS name space) and to collect this data over long periods of time potentially spanning multiple years.

This paper focuses on the challenges of achieving this goal by answering the following main research question: "*How can one perform a daily active DNS measurement of a significant proportion of all domains on the Internet?*". The main contributions of the paper are that we show how to:

- Scale such a measurement to cope with the largest TLD (`.com` with 123M names).
- Ensure that the traffic such a measurement generates does not adversely affect the global DNS infrastructure.
- Efficiently store and analyse the collected data.

Our measurements create a novel large-scale dataset of great value to the research community as well as in other contexts (e.g. for security and forensic purposes). Our ultimate goal therefore is to make the data accessible to others. How we will do this is discussed at the end of the paper.

Finally, in order to validate our system in practice and to illustrate potential uses of the data it collects, we performed two case studies. Given the growing research interest in cloud services, the case studies focus on the use of cloud e-mail services. Based on ten months of data collected by the measurement system between March 2015 and January 2016, we studied the following questions:

- Is Google the most popular cloud mail service provider, or are others, such as Microsoft or Yahoo, more popular?
- Which of these three providers sees the fastest growth?
- Do domains that use these cloud mail services use the Sender Policy Framework (SPF) [1] to combat e-mail forgery, especially since most providers support SPF?

Structure of this paper – Section II introduces our long-term research goals and the challenges that achieving these

goals pose. Section III discusses and motivates design choices and the resulting design of the measurement system we created. Section IV examines operational experiences with the measurement system, and analyses the impact the system has on the global DNS infrastructure. Section V presents two case studies, which serve both to validate the system, and to illustrate the value of the collected data. Section VI covers background information and related work. Section VII describes how we intend to make result data accessible to the academic research community. Finally, in Section VIII, we present our conclusions and discuss future work.

II. Goals and Challenges

A. Goals

Our research goal is to create a large-scale data set covering the state of the DNS for a significant proportion of the global name space. The data set should record this state at regular intervals, in order to be able to create time series tracking trends and developments on the Internet. To achieve this ultimate goal, we define the following sub-goals:

G1 **Measure every single domain in a top-level domain (TLD)** – this allows us to build a comprehensive picture of large parts of the DNS name space.

G2 **Be able to measure even the largest TLD (`.com`)** – if the system is capable of measuring `.com` (123M names) it can also measure other, smaller TLDs.

G3 **Measure a fixed set of relevant resource records for each domain** – the DNS has different resource record types that serve specific purposes. In Table I we define the set of queries we want to perform. Queries have been chosen such that they cover the most common DNS uses with the minimum number of queries.

G4 **Measure each domain once per day** – to be able to create reliable time series, each domain must be measured exactly once every 24 hours.

G5 **Store at least one year's worth of data** – to do meaningful research, we should be able to store data that covers at least one year, and preferably a longer period.

G6 **Analyse data efficiently** – we expect to be collecting data for tens of millions of domains; this means that we must explicitly design for efficient analysis through modern technologies such as the Hadoop ecosystem.

G7 **Scalability** – the measurement should scale to both handle TLD growth and to measure additional TLDs. We initially foresee measuring the main generic TLDs (gTLDs) `.com`, `.net`, and `.org`, as together these contain 50% of domain registrations in the global DNS.

B. Challenges

To meet the goals above, a number of challenges will have to be overcome. These challenges are outlined below:

C1 **Query volume** – as *G1* and *G4* state, we want to be able to measure all domains in the largest TLD (`.com` with 123M names) once every 24 hours. For each name, 14 queries are performed (*G3*). Next to direct queries, the system needs to send additional queries as part of normal DNS recursion (e.g. to find the authoritative name servers for a domain). A conservative estimate is that this requires one additional query per domain. Thus, querying every domain in `.com` requires at least 1.85B queries per day.

C2 **Query pacing** – a challenge related to *C1* is pacing of queries. It is important that the queries we send do not impose an excessive load on authoritative name servers. Especially traffic flows to the top-level servers that are authoritative for the TLDs that are measured need to be monitored, as queries for individual domains also lead to queries to these servers due to the hierarchical way the DNS is organised. Similarly, we have to monitor the traffic volume to large hosting providers, since these may provide authoritative DNS services for large numbers of domains.

C3 **Storage** – taking the `.com` TLD as yardstick – and assuming that each of the 14 queries performed for each domain returns ±150 bytes of data – more than 240GB of results need to be stored per day for `.com` alone. Considering *G5* and *G6* this is particularly challenging.

C4 **Robustness** – the measurement must run continuously and not suffer from downtime due to maintenance or crashes.

C5 **Ease of operation** – to meet most of the other challenges outlined above, we foresee a distributed system of machines that perform the measurement. Management and administration of such a distributed infrastructure has to be simple. Additionally, scaling the measurement to incorporate more TLDs should also be straightforward.

Resource Record	Description
`SOA`	The Start Of Authority record specifies key parameters for the DNS zone that reflect operational practices of the DNS operator.
`A`*	Specifies the IPv4 address for a name.*
`AAAA`*	Specifies the IPv6 address for a name.*
`NS`	Specifies the names of the authoritative name servers for a domain.
`MX`	Specifies the names of the hosts that handle e-mail for a domain.
`TXT`	Contains arbitrary text strings. This record type is used to convey – among other things – information required for spam filtering and is also often used to prove control over a domain to e.g. cloud and certificate authorities.
`SPF`	Specifies spam filtering information for a domain. Note that this record type was deprecated in 2014 (RFC 7208), we query it to study decline of an obsolete record type over time.
`DS`	The Delegation Signer record references a `DNSKEY` using a cryptographic hash. It is part of the delegation in a parent zone, together with the `NS` and establishes the chain of trust from parent to child DNS zones in DNSSEC.
`DNSKEY`	Specifies public keys for validating DNSSEC signatures in the DNS zone.
`NSEC(3)`†	Used in DNSSEC to provide authenticated denial-of-existence, i.e. to cryptographically prove that a queried name and record type do not exist.†

*Query is performed for the apex, 'www' and 'mail' labels, e.g. for `example.com`, `www.example.com` and `mail.example.com`.
†Query is only performed for DNSSEC-signed domains with one or more `DNSKEY` RRs.

TABLE I
Query types to perform

III. Measurement System Design

We have designed and implemented a measurement system to meet the goals and challenges discussed in Section II. This section takes a detailed look at the design of this system. The section is divided into two parts. The first part discusses and motivates the major design decisions taken while creating the measurement system. The second part describes the resulting system design and its implementation.

A. Design Choices

Before setting out to design and implement such a large-scale measurement system, we carefully considered key

choices to make in order to ensure that the system tackles all the challenges and meets all the goals discussed in Section II. This subsection discusses the major design decisions made while creating the measurement system and motivates our choices by discussing the options we explored.

1) DNS software: Given the goal of the system, the most important decision to be made concerned the software to use to perform the actual DNS queries that make up the measurement. Two options were considered:

A bare metal approach – in a bare metal approach the focus is on maximum measurement speed. An example of this approach is the ZMap network scanner [2], which performs network scans by directly generating Ethernet frames. This approach bypasses all intermediate layers in the network stack, allowing scans at near line-speed. While a bare metal approach is a potentially attractive way to tackle the measurement speed challenge we face, there are disadvantages to taking such an approach. Most importantly, resolving DNS queries is a complex task, much more complex than e.g. the simple port scans ZMap performs. Re-implementing DNS resolution in a bare metal fashion would require significant effort and runs a high risk of bugs that adversely affect the reliability of the measurement system (challenge *C4*).

Using off-the-shelf DNS software – this option relies on maximum re-use of existing software. The measurement software would need to incorporate a simple DNS stub resolver that is capable of sending single queries, and the more complex task of DNS recursion is left to an off-the-shelf implementation. The advantage of such an approach is that it entails the smallest risk of falling into the pitfalls of the complex task of implementing DNS recursion. The disadvantage is, of course, that such an approach will be slower.

Taking the advantages and disadvantages of these two options into consideration, we chose to explore the second option – using standard DNS software as much as possible – in more detail. Firstly, this approach requires the least complexity in terms of software development. This is important especially because it provides the best guarantees for the robustness of the system (challenge *C4*) which is a key requirement for long-term data collection (goal *G5*). Secondly, our intuition was that this option would perform sufficiently well to meet challenge *C1*. Next to that, the top-speed performance offered by the first option (bare metal) is not actually a requirement. Rather, to manage the impact of the measurement on the global DNS (challenge *C2*) there must be a trade-off between speed and impact of the measurement. In order to confirm our intuition that this approach performs sufficiently, a proof-of-concept was implemented, the goal of which was to measure the medium-sized `.org` top-level domain. Given the time taken to measure this TLD, we could extrapolate that this approach would make meeting challenge *C1* (measuring very large TLDs such as `.com`) feasible. Based on these considerations we chose to proceed with the implementation of the second approach. All that remained was to determine the impact of the measurement on the global DNS; this is discussed in Section IV-B.

2) Scalability of the measurement: The second design decision focused on how to best scale the measurement system (goal *G7* and challenge *C5*). The first option considered was to run the measurement software on a single system. Given measurements we performed for an earlier study [3], we knew from experience that this would put high requirements on the system on which the measurement would run, mainly in terms of CPU utilisation. Choosing this option would therefore make it hard to scale the measurement in the future.

To ensure scalability, we thus chose a distributed approach with a central orchestration system and a swarm of worker nodes. Given the ready availability of cloud computing stacks, we focused on an implementation that is amenable to deployment on cloud platforms, and chose to implement worker nodes as a virtual machine image. While we initially envisaged a deployment of the measurement system in a single location, this design choice means that we can scale up to commercial cloud platforms if we run out of local resources, and it also means that we can relocate parts of the measurement to other geographical regions. The latter may be advantageous for measurements on, for example, country-code top-level domains (ccTLDs) with a strong geographic binding, where measuring from a local vantage point relative to the ccTLD can have performance benefits in terms of network latency.

3) Data format and analysis: The final design considerations concern storage and analysis of the measurement results (goals *G5* & *G6* and challenge *C3*). The de-facto toolchain for analysing big datasets – such as the one our measurement system collects – is the Hadoop ecosystem[1]. Thus, we designed the system such that the resulting measurement data is suited to processing in the Hadoop ecosystem.

For storage, we decided on a two-tiered approach. In the first step, results are stored in the Apache Avro file format[2]. Avro is a structured, self-describing data serialisation format with built-in support for compression, which is used by the system to reduce the storage size and thus meet *C3*. We use a simple flat schema that encodes a single DNS record as one row, with sparse storage. This means that only fields belonging to the particular DNS record type that is being stored are filled, other fields are assigned a `null` value. This approach was selected over nested structures because it is simple to map to most database paradigms. As a second step, to further improve analysis performance, measurement data is converted to the Parquet[3] columnar storage format *in situ* on a Hadoop cluster on which data is analysed. Traditional row-oriented databases are optimised for access to all the data in a single row. Queries that aggregate data from many rows and that, for instance, accumulate counts based on filters on certain columns are typically inefficient on this type of database. A columnar storage system stores all data in a single column sequentially. This makes aggregation across a single or a few columns much more efficient. Additionally, because data in a single column uses the same data type and is typically made up of similar values, sequential columnar data can be compressed efficiently using e.g. run-length and delta encoding techniques.

There are two reasons for this two-tiered approach. First, storing measurement results in the Avro format with the

[1]For an in-depth introduction to Hadoop, see [4].
[2]http://avro.apache.org/
[3]http://parquet.apache.org/

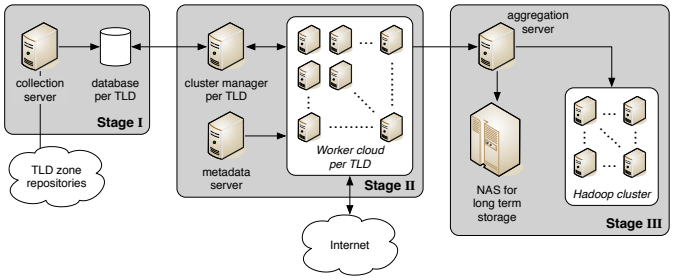

Fig. 1. High-Level Architecture

schema discussed above makes this data suitable for separate long term archival (see III-B3 below). The row-oriented nature of the Avro schema means that the data can easily be converted to future database paradigms. Second, the Avro files are structured such that they can also be analysed outside of a Hadoop cluster. All results relating to a single domain name are stored sequentially in an Avro file. Knowledge of this structure allows for development of efficient analysis tools without the help of the Hadoop ecosystem. While performance will be less than on a Hadoop cluster, this makes the data usable to researchers who do not have access to such resources.

To analyse the data collected by the measurement system, we use the Apache Impala[4] engine. This allows us to perform batch-based analyses using SQL queries. The optimal batch size depends on the complexity of the query; in general, processing is done in batches per day or per calendar month. As an example, the analyses we performed for the case studies discussed in Section V took under 2 hours each, processing over 511 billion data points. In the future, we will explore additional technologies, such as Apache Spark[5], which allows for streaming processing as the measurement data comes in.

B. System Design and Implementation

Given the design considerations discussed in the previous subsection and given the goals and challenges outlined in Section II-B we arrived at the design as depicted in Figure 1. The figure shows an overview of the entire system and identifies each of the three stages the system is divided into with a grey rectangle. Each stage is described in detail below.

1) Stage I - Input Data Collection: Stage I collects input data, consisting of full DNS zones for the TLDs measured (Table II). New zone data is collected once per day, after which a daily delta is computed (domains added and removed). The domain names in a TLD are stored in a separate database per TLD. Each database has two tables, one for the set of active domains (i.e. the current state of the DNS zone), and one with all domains seen since the start of the measurement. The latter table reflects developments in the zone and stores timestamps for when a domain name was first seen, when it was last removed from the zone, and when it reappeared in the zone (the latter two are only present if applicable). This design decision means that the Stage I database can be used for stand-alone analysis of changes in the TLD zones. This makes some forms of analysis more efficient, which contributes to achieving goal *G6*.

2) Stage II - Measurement: The second stage has three functional components. The first is a cluster manager that takes care of dividing work across the second component, a cloud of worker nodes. The third component is a metadata server. It maintains up-to-date IP address to autonomous system (IP-to-AS) mappings as well as Geo IP data.

The cluster manager collects chunks of work from the database. A chunk consists of a set of domains that were last measured before midnight UTC. This ensures that each domain is queried exactly once per day (goals *G1*, *G4*). Chunks are added to a pool of work to be performed, and the domains in each chunk are marked as checked out in the database. As workers process chunks of work, the cluster manager takes care of administrative tasks, managing the pool of available work, and updating the database upon job completion by workers. It also monitors measurement progress and will reassign a chunk to a new worker if its current worker takes too long. This prevents worker crashes from causing parts of the measurement to fail (challenge *C4*).

Worker nodes obtain chunks of work from the cluster manager, perform the queries specified in Table I for each domain in the chunk and collect the results. Workers store all resource records included in the answer section of the DNS response, including all DNSSEC signatures, `CNAME` records and full `CNAME` expansions. Upon completion of a chunk, the worker reports back to the cluster manager and obtains new work. The worker also enriches the collected data based on available metadata (IP-to-AS and Geo IP) and submits the measurement results to the storage system in Stage III. Finally, a worker node will check in with the metadata server to obtain new metadata if available. Worker nodes are generic components; this helps meet goal *G7* as additional workers can be deployed easily to increase measurement throughput.

The cluster manager and worker software were custom-developed (in C) for this measurement system. The worker uses LDNS[6] for all DNS-specific processing (issuing queries and parsing query results). To reach goal *G2* and challenge *C1*, workers run multiple query threads. This prevents workers from prolonged inactivity if queries time out (which may halt a querying thread for up to 30 seconds). Each worker node also runs a local DNS resolver for which we selected Unbound[7] as software. Caching by this resolver helps reduce the query load on the global DNS (challenge *C2*). Caching of infrastructural information, such as the IP addresses of authoritative name servers, is particularly useful, as large numbers of domains run by a single operator tend to share the same authoritative servers. To ensure fresh data is collected each day, the resolver caches are configured to expire every day. In addition to caching, another important function of the DNS resolver is distributing queries evenly over authoritative name servers, which is especially important to reduce the load on top-level domain servers. Unbound strikes a good balance between query round-trip time (RTT) and distribution of queries over

[4]http://impala.io/
[5]http://spark.apache.org/
[6]https://www.nlnetlabs.nl/projects/ldns/
[7]http://unbound.net/

TLD	Registry	#domains	(% of DNS)	Stage I time (Mar-Dec 2015) mean	σ
.com	Verisign	123.1M	(41.2%)	4h 17 min.	1h 15 min.
.net	Verisign	15.6M	(5.2%)	45 min.	31 min.
.org	PIR	10.9M	(3.6%)	19 min.	6 min.
total		149.6M	(50.0%)	5h 20 min.	1h 20 min.

TABLE II
INPUT ZONE CHARACTERISTICS

TLD	#worker VMs	averages over Mar-Dec 2015 time (batch) mean	σ	time (total) mean	σ
.com	80	54 min.	6 min.	17h 10 min.	2h 23 min.
.net	10	52 min.	8 min.	14h 29 min.	2h 15 min.
.org	10	37 min.	4 min.	7h 19 min.	57 min.

TABLE III
STAGE II MEASUREMENT DURATION

multiple authoritative name servers by randomly selecting authoritative name servers with an RTT below 400ms [5]. As Section IV-B will show, this results in a good distribution of queries over top-level domain servers.

Finally, as discussed in Section III-A, Stage II of the measurement system is based on virtual machines. These currently run on top of a private cloud infrastructure based on OpenStack[8]. While we run a large number of worker nodes, as will be discussed in the next section (IV), these consume minimal resources. In the current setup, for each worker only a single CPU core, 2GB of RAM and 5GB of disk are allocated.

3) Stage III - Storage and Analysis: Stage III takes care of two tasks. First, data is copied from the aggregation point, where workers deposit data, to long-term storage. This serves two purposes: safeguarding a backup copy of the data on reliable storage redundantly distributed over two locations, and retaining the unmodified source data as measured. Second, data is copied onto a dedicated Hadoop cluster, which we use for analysis. During the copying process, the data is converted to the Parquet format discussed in Section III-A3 above to enable efficient analysis of the data later on.

IV. OPERATIONAL EXPERIENCES

This section discusses operational experiences with the measurement system. It covers what data we currently collect, provides performance metrics, and discusses the impact of the measurement on the global DNS infrastructure. Stage I of the measurement system became operational in July 2014, while Stages II & III have been operational since February 2015.

We obtained access to the zone files of the .com, .net and .org generic top-level domains. Access to these zone files is regulated under contracts[9,10] with the registry operators of these TLDs. Table II lists the characteristics of each zone.

A. Performance

Stage I retrieves each TLD zone twice a day, extracts the list of domain names from each TLD zone, and computes the delta relative to the previous version. It then updates the databases for each TLD. The rightmost columns of Table II show the average running times for Stage I over 2015 as well as the standard deviation. The variability in running times for .com and .net is caused by intermittent throttling of the zone file download by registry operators. Stage I runs are scheduled to complete before the cluster manager starts checking out batches of work for Stage II. The two daily runs along this schedule guarantee that new domains are part of the measurement within 24 hours of appearing in a TLD.

Table III shows the configuration and average measurement times for Stage II over the period March-December 2015. The table shows the number of workers per domain, the average measurement time per batch, and the total duration of a single day measurement. For the latter two values, the mean as well as the standard deviation is displayed. As shown, the average measurement time per batch varies significantly between TLDs. Closer examination reveals two reasons for this. For .com, the higher average duration is due to certain batches being dominated by domains registered from China. Per query network latency causes these batches to have significantly longer measurement times. The average round-trip time (RTT) per query for these batches is up to 7 times higher than average RTT. For .net, the duration per batch is higher because the RTT for queries is about a third higher than for .org on average. There appears to be no discernible cause for this; it is most likely due to a difference in the infrastructure of the TLD. As Table III shows, the system manages to perform a full measurement well within a 24-hour window, meeting goals *G1*, *G2* and *G4*. The total measurement time per day, however, varies substantially. This can be explained by two effects. First, stage I runs twice per day in order to ensure that new domains become part of a measurement within 24 hours. As a result of this, there are occasional measurements for a small number of batches at the end of the day. The total measurement time is computed as the time between the first measured domain on a day and the last. Thus, these late night batches skew the total measurement time. Second, all three TLDs included in the measurement have grown over the period for which the value was computed, leading to longer overall measurement times at the end of the period. Nevertheless, even the longest measurement (for .com) has ample room to run longer while still remaining within a 24-hour window. Furthermore, during the initial tuning of the system, we experimented with the number of workers per domain to bring down the measurement time. There is a strong relation between the number of workers and the average duration of the measurement, despite the fact that worker VMs share hardware and network infrastructure. We can thus meet goal *G7* and cope with growth in the number of domains by adding additional workers. The average batch duration and overall measurement time are monitored continuously so additional workers can be provisioned on time to remain within the 24 hour window set in goal *G4*. CPU utilisation of the workers is also monitored and we aim for an average utilisation between 25% and 50%, to strike a balance between keeping room for brief bursts of high activity while not underutilising resources. In general, around a quarter of

[8]http://www.openstack.org/

[9]For .com and .net see http://www.verisigninc.com/en_US/channel-resources/domain-registry-products/zone-file/index.xhtml

[10]For .org see http://pir.org/resources/file-zone-access/

	results for December 31, 2015				averages over Mar-Dec 2015			
					results/domain		failed domains	
TLD	#results	#domains	size	(uncompressed)	mean	σ	mean	σ
.com	1419M	122.3M	28.8GB	(211.6GB)	11.75	0.07	0.83%	0.17%
.net	166M	15.5M	3.4GB	(24.3GB)	11.05	0.15	1.21%	0.19%
.org	125M	10.7M	2.5GB	(18.4GB)	11.77	0.09	1.60%	0.22%
total	1709M	148.5M	34.8GB	(254.3GB)	11.68	0.08	0.92%	0.17%

TABLE IV
STAGE II RESULT STATISTICS

CPU use on the worker is due to our measurement application while the other three quarters are used by Unbound.

Table IV gives an overview of daily results. The left-hand side of the table shows the statistics for December 31, 2015. The first column shows the total number of results per TLD, followed by the number of domains for which data was successfully collected. The next two columns show the size of the collected data per TLD. The right-hand side of the table shows two average metrics over the period March-December 2015. These metrics are an indication of the stability of the measurement. Both metrics vary only slightly over the ten-month period. The first metric is the average number of results per domain. As the table shows, this number is lower than the 14 queries performed for each domain (Table I). There are two reasons for this. First, only data in the answer section of a DNS response is recorded. If the name exists but no record of the queried type exists for this name, the server will return a response with an empty answer section (a `NODATA` answer). Second, results for queries that failed with a response code other than `NOERROR` (the query succeeded) or `NXDOMAIN` (the queried name does not exist) are discarded. If another response code is returned, further queries for the domain are aborted to prevent workers from stalling on misconfigured domains. The second metric is the average percentage of domains for which no data could be obtained. As shown, only about 0.9% of domains fail to return any results to queries; the name servers for these domains are either misconfigured, or the domains are so-called lame delegations (domains for which none of the delegated name servers respond to queries). There appears to be a downward trend in the number of failures over the current measurement period, indicating that more queries succeed. Finally, looking at the amount of data produced per day shows that the data compression discussed in Section III-A3 works well, achieving a stable average compression rate of 1 : 7.4. From the start of the measurement in February 2015, the system has collected over 10TB of compressed data. As our current setup can store up to 50TB of data, goal *G5* is also met.

B. Impact on the DNS

As discussed in Section II-B (challenge *C2*), we have to ensure that the measurement does not impose an unacceptable burden on the global DNS infrastructure. There are two reasons for this. First, we consider it ethically unacceptable if the measurement were to put significant load on individual DNS servers. This might negatively impact DNS performance for 'real' users. Second, the contracts under which we gained access to the TLD zone files for `.com`, `.net` and `.org` all stipulate that it is not allowed to run *"...high volume, automated, electronic processes that send queries or data to the systems of [the] Registry Operator ... except as reasonably necessary..."*. While this clause pertains mostly to the registry service itself and is in spirit intended to stop aggressive scanning of registry data in order to claim domain names that have also been registered in other TLDs, we nevertheless also apply it to our measurement and strive to minimise the load on the DNS servers operated by the TLD registries.

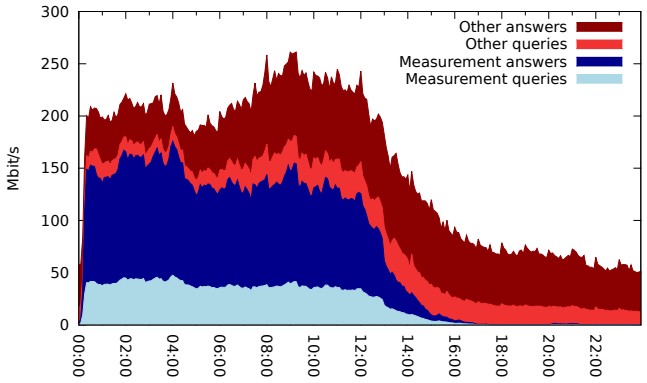

Fig. 2. Measurement flows versus other flows from the SURFnet network

An obvious way of limiting the load imposed by the measurement is to actively rate limit queries. We chose not to do this for two reasons. First, to support this form of throttling, modifications to the standard DNS resolver software we use would be required. Second, the query load is not distributed evenly over authoritative name servers because of the hierarchical nature of the DNS. Servers higher up the DNS hierarchy, i.e. authoritative name servers for top-level domains, typically receive many more queries because they have to be consulted to find the specific authoritative name servers for every domain name measured. Conversely, these servers higher up the DNS hierarchy are designed and configured to handle many more queries. Thus, we would have to apply a different rate limiting policy to these servers, making the measurement system much more complex. Instead, our approach is to analyse the impact of the measurement, to show that our system design makes rate limiting unnecessary.

To gauge the impact of measurements, flow data was collected for the network from which the measurement operates. The infrastructure is hosted by SURFnet[11], which collects sampled flow data from its core routers with a sampling rate of 1 : 100. While sampling means some flows (especially very small ones) will be missed, it provides a good picture of the top talkers. We are interested in these, since they are the systems on which the highest query burden is imposed. To get a feeling for the query volume that the measurement generates, we compared the query volume to that of the entire SURFnet network. Figure 2 shows this comparison; the traffic volume from the measurement system exceeds the regular DNS traffic from the SURFnet network. This network has over 1 million end users in some 180 institutes for higher education and research on it, so the query volume generated by the measurement system is quite significant.

[11]The National Research and Education Network in the Netherlands

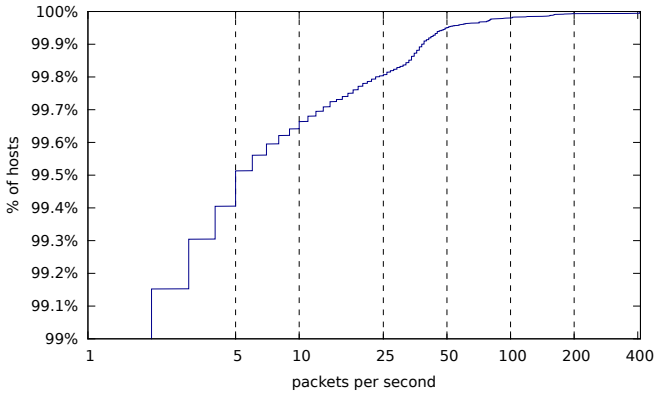

Fig. 3. CDF showing the distribution of flow rates to individual IPs

To quantify how much traffic individual IP addresses receive, we examined outgoing flows for 24 hours ordered by average number of packets per second (pps)[12]. Figure 3 shows the top 1% of a CDF for the flow rate in pps. What is immediately evident is that there are very few flows with a high pps rate. Second, no flow exceeds 400 pps. Only 35 IPs are true top talkers (more than 100 pps). Unsurprisingly, the top of the list consists exclusively of gTLD DNS servers for `.com` and `.net`. On average, each of these servers (there are 13) receives ±400 queries per second. A study from 2011 [6] reports that one particular gTLD DNS server receives over 900 million queries per day (±10,400 per second). Under the conservative assumption that the query load did not increase since 2011, our measurement would add 3.8% to the query load of that server. More recent figures from Verisign[13] suggest that the actual figure is probably lower. Given that the measurement generates some 2 billion queries per day, this would account for between 0.3% and 1.6% of all queries. Also, in private communication, Verisign has indicated that they see the measurement and that while it is a non-trivial amount of traffic, it is not problematic. The next group of top talkers receives less than 200 queries per second. Closer examination shows all of these belong to companies that practice domain parking[14]. While we have no data on the infrastructure of these companies, it is safe to assume that a query rate of less than 200 queries per second can easily be handled by a name server. One thing should be noted: the figures provided are averages over one measurement period, meaning there may be peaks during which more traffic is sent. While it is hard to quantify to what extent such peaks occur, they are most likely not extreme as that would have showed up in Figure 2.

This analysis demonstrates that the measurement does not impose an excessive burden on the global DNS infrastructure (challenge *C2*). Nevertheless, the load is significant, which makes it undesirable that large numbers of researchers start running similar measurements. Therefore, we pay specific attention to data sharing in Section VII.

[12]The flow rate was adjusted to correct for the 1 : 100 sampling.
[13]http://www.verisign.com/assets/infographic-dnib-Q32015.pdf
[14]https://en.wikipedia.org/wiki/Domain_parking

V. Case Studies

This section contains two case studies that cover the questions regarding the use of cloud mail service providers introduced in Section I. These serve to validate the results our measurement system produces and to demonstrate how measuring the DNS can be a valuable instrument that provides insight into operational practices on the Internet.

A. The growing use of cloud e-mail service providers

E-mail is one of the oldest services on the Internet. Where up until the mid 2000s mail was either hosted on premises or a service provided by the ISP, there is nowadays a trend to outsource e-mail to cloud service providers. In this context, we discern three classes of service provider. First, *hosting providers* offer domain registration, web hosting, (virtual) private servers and e-mail. These providers often provide basic e-mail services with few user mailboxes or the option to forward mail to an address set by the user. Second, *cloud providers* offer fully hosted office ICT services. Their service offering in the e-mail space is often rich, allowing customers to provision e-mail accounts for all their users and integrating every day office requirements like calendaring and document sharing and editing. Third, *protection services* focus on protecting e-mail against malware, phishing, spam and other malicious activity. They process e-mail to filter unwanted content and forward sanitised results to another mail service. This case study focuses on the second category, *cloud providers*. Based on data collected by our measurement platform over the ten-month period between March and December of 2015, we study the use of such services in the `.com` top-level domain.

To identify which e-mail providers handle e-mail for the most domains in the `.com` TLD, we examined the *Mail eXchanger* (`MX`) records in the DNS. The first step in the analysis identified the top `MX` records used by domains in `.com` by examining all data points for a single day (March 1, 2015) in the data set. `MX` records were grouped by second-level domain (SLD) to filter out multiple records that point to hosts within the same service provider. For example, the SLD for Microsoft's Office 365 cloud service offering is `outlook.com`. We manually classified the results of this analysis to determine which service provider the `MX` records belong to and in which of the three classes of service provider they fall. Looking at cloud providers, we find that on March 1, 2015 the top three consists of what we would term the usual suspects: Google (serving 4.09M domains), Microsoft Office 365 (948k domains) and Yahoo (609k domains). Note that, while these are large numbers, cloud providers are not the dominant mail handler. The most common `MX` record (±27M) by far for domains in `.com` points to GoDaddy, a large domain name registrar and hosting provider.

In the introduction to this paper we asked the question: "*which cloud e-mail provider sees the fastest growth?*". Intuitively one might answer "Google". Surprisingly, however, that is not the case. Both in absolute numbers as well as in relative growth, Microsoft grows the fastest between March and the end of December 2015. In absolute numbers, Microsoft went from 948k domains using their service to 1.44M, Google

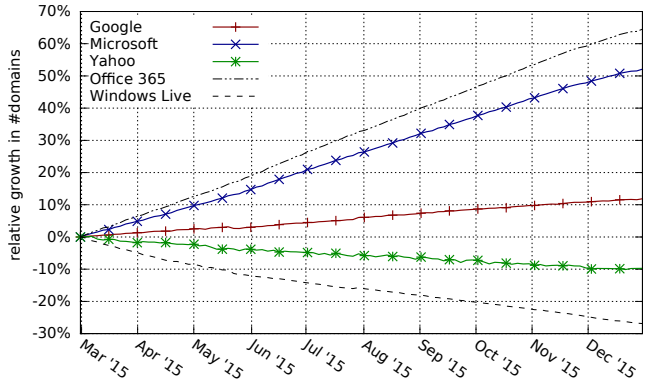

Fig. 4. Relative growth in use of cloud e-mail providers for the .com TLD

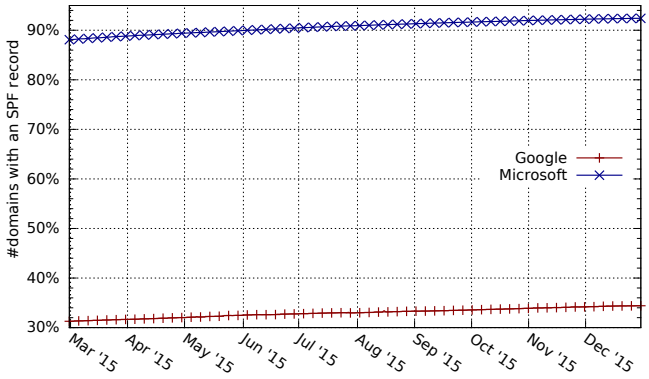

Fig. 5. SPF usage growth for cloud e-mail providers in the .com TLD

went from 4.09M to 4.57M, and Yahoo dropped from 609k to 549k. Figure 4 shows the growth relative to the starting point of the analysis (March 1, 2015) in the number of domains that use one of the three cloud mail providers. Again, Microsoft is by far the fastest grower. However, there is a twist to that figure. The blue line shown for Microsoft is an aggregate of domains that use Windows Live (formerly Hotmail) and Office 365. Microsoft has discontinued Windows Live as a brand for mail services and this is visible in the data. The dashed line shows the decline in use of Windows Live (`hotmail.com`). Looking purely at Office 365 (dashed-and-dotted line), Microsoft's growth is even more noticeable. One explanation for Microsoft's fast growth can be that the large registrar and hoster GoDaddy (mentioned above) is an Office 365 reseller since 2014[15]. A staggering 74% of the growth in number of domains using Office 365 can be directly attributed to domains registered through GoDaddy. Also of interest is the slow decline of Yahoo. While we did not look into this in detail, we note that Yahoo has regularly been in news headlines over the past two years as struggling.

The measurement system is not only suited to one-shot analyses and time series, but can also be used to detect significant anomalies in the DNS name space. To illustrate this, we discuss an example anomaly encountered while performing the analysis of `MX` records above. In the middle of May 2015 a sharp decline occurred for one of the top `MX` SLDs, from 2.51M domains advertising this record to 1.27M. While the provider the `MX` SLD belongs to is not a cloud mail provider[16], we investigated the drop nevertheless, to ensure that this anomaly was not caused by problems with the measurement. Interestingly, it turns out that this `MX` SLD is associated with a service that appears to be targeted at companies specialised in domain parking[14]. The goal of the service is to respond to e-mails sent to parked domains. The assumption behind this appears to be that users may erroneously send e-mail to parked domains; rather than returning a standard error message, the service will return a customised error containing advertisements. The sharp drop in May is caused by a mass change in `MX` records previously pointing to this service. We did not analyse the rationale for this change further, but leave this to future study.

B. Sender Policy Framework (SPF) practices

A common problem with e-mail is illegitimate sending of e-mails that seemingly originate from a certain domain but are in fact sent by a rogue or compromised mail server with no relation to that domain. To combat this, the Sender Policy Framework (SPF, standardised in RFC 7208 [1]) was introduced. SPF allows domain owners to specify which servers may send e-mails on their behalf, and as such helps combat forgery. Domain owners publish SPF information[17] in the DNS by means of a `TXT` record (cf. Table I). This case study evaluates the use of SPF by domains that use one of the three large cloud e-mail providers from the previous case study.

Like the first case study, data was analysed over a ten-month period to determine the presence of SPF information for domains that use either Google, Microsoft, or Yahoo to handle their e-mail. Figure 5 shows the result of this analysis. The lines in the figure represent the fraction of domains that use either Google or Microsoft and that publish SPF information in the DNS. Yahoo is not shown in the figure as less than 0.4% of domains that use Yahoo's mail service publish SPF information. Significant numbers of both users of Google's as well as of Microsoft's services publish SPF records. There is, however, a surprising difference between the two. As the figure shows, around 31.3% of domains that use Google publish SPF information, growing to 34.4% at the end of the period. For Microsoft these figures are significantly higher, growing from 88.1% to 92.4%. Both Microsoft and Google provide instructions on how to publish SPF information when using their service. We have not examined in detail why this difference in SPF deployment occurs. One possible explanation is that the majority of domains that use Microsoft's Office 365 do so via resellers that set the appropriate SPF records automatically. For example, of domains using Office 365 that are registered through GoDaddy, 98.8% publish SPF information. This is

[15] http://www.computerworld.com/article/2487663/enterprise-applications/godaddy-touts-simplicity-over-price-as-it-launches-office-365-sales.html

[16] For ethical reasons, we do not disclose the name of the company as it is not a large publicly traded company.

[17] http://www.openspf.org/SPF_Record_Syntax

certainly worthwhile of further study as the use of SPF is an important tool in combating e-mail fraud.

VI. Background and Related Work

Measuring the DNS has a number of dimensions. In particular, we identify the following: the measurement goal, passive and active approaches, "one-shot" versus measurements over time and vantage points of the measurement. We note that these dimensions are not necessarily independent; for instance: in most cases passively collecting DNS data only makes sense if the measurement is distributed, while collecting data at authoritative name servers is probably limited to a few vantage points as it is difficult for researchers to gain access to such data sources. In the next subsections we describe these dimensions and discuss past and present research efforts in measuring the DNS in the context of these dimensions.

A. Goal of the measurement

The DNS can be measured to study the behaviour of the DNS infrastructure itself (e.g. security, resilience, ...), or it can be measured because it provides information about operational practices on the Internet (for example the presence of `AAAA` records says something about IPv6 deployment). A notable example (on account of scale and running time) of studying the DNS itself is the Internet Domain Survey [7]. This automated survey publishes statistics on the number of IP addresses that have a name associated with it in reverse DNS and has been running since 1987. Another example is a study by Osterweil et al. [6] that examines the day-to-day performance of one of the authoritative name servers for the `.com` and `.net` TLDs. Pappas et al. [8] study the effect of configuration errors on the DNS; notably, they perform a number of one-shot active measurements that sample around 10% of the domains in the `.com` TLD.

Examples of studies examining the DNS to uncover underlying behaviour of, or on, the Internet can, e.g., be found in the security space. Works by Bilge et al. [9] and Perdisci et al. [10] study malicious domain names and botnets, respectively.

B. Passive versus active measurements

The most well-known method for passive DNS measurements is *passive DNS* (pDNS) [11]. In most cases, pDNS is used to capture DNS traffic between a recursive caching name server (resolver) and the authoritative name servers it communicates with. This ensures that the privacy of users of the resolvers where data is captured is preserved. There are large scale deployments of pDNS that capture data at many vantage points. Notable examples are Farsight Security's DNSDB[18] and the pDNS infrastructure operated by CERT.at[19]. These large pDNS deployments are often used in operational security contexts. They are commonly operated by or for Computer Security Incident Response Teams (CSIRTs). Research that relies on pDNS often focuses on security (e.g. the two papers discussed in the previous subsection [9], [10]). In addition, pDNS is also used to study operational aspects of the DNS. In this case pDNS is often deployed at specific vantage points, for example, [12] and [13] study DNS traffic for the `.nl` and `.it` TLDs respectively. Finally, pDNS data can be used to enhance other network measurements. For example, Bermudez et al. [14] use DNS data to tag network flow data.

In contrast, active measurements, such as the system we introduce in this paper, work by sending targeted queries to the DNS. There are fewer examples of active DNS measurements in the literature. Examples include work by Schomp et al. [15] who use active scans to investigate the client-side DNS infrastructure. They perform these scans by randomly selecting IPv4 addresses and address blocks to find certain types of DNS servers. Their goal is to characterise the behaviour of the DNS servers themselves, not to collect DNS content. Earlier work by the authors of this paper [3], [16] used active DNS measurements to study aspects of the DNSSEC protocol. Zhu et al. [17] study the deployment of DNS-based Authentication of Named Entities (DANE) by actively sending DNS queries for all DNSSEC-signed domains in `.com` and `.net`.

When compared to existing work that uses active measurements, the approach taken in this paper stands out in two ways. First, our approach is generic, that is: not specifically designed to study a single aspect of the DNS or the Internet. Second, the scale at which we measure is orders of magnitude larger than previous studies that use active DNS measurements.

Current passive DNS deployments, such as the aforementioned DNSDB and CERT.at systems, are comparable in scale to our active measurement approach. Where pDNS differs from our approach is that pDNS systems collect dynamic DNS data that is the result of queries by end clients. Thus, pDNS databases will typically contain information on domain names that are actively queried by clients and will contain more data if domains are more popular. The spread of TLDs covered by a large scale pDNS setup will typically be very diverse. In contrast, our active measurement covers DNS data for all domains (also domains that are unpopular) in the TLDs we measure, and it has data for each of these domains for every day. Thus, our approach is complementary to pDNS.

C. Time

For certain research, it is sufficient to perform one or perhaps a few single shot DNS measurements. This is the case, for example, for the studies in [3], [13], [16], [18], [19], [20]. All of these are based on "one-shot" measurements. Other research, however, looks at developments over time and thus needs DNS data collected over a period of time. For instance, [9], [10] use pDNS data collected over longer periods. There are also examples of active measurements that cover longer periods, e.g. [7], [17]. The intervals at which data is collected varies. For pDNS, data points are scattered over time, as they depend on live queries that arrive at unpredictable times. For active measurements, this varies from twice per year [7] to daily in case of the approach taken in this paper, and by [17].

D. Vantage points

The final dimension is whether just a single or multiple vantage points are used to perform the measurement. Whether

[18] https://www.dnsdb.info/

[19] The Austrian National CERT team, http://www.cert.at/index_en.html

or not multiple vantage points are necessary depends on the measurement. For instance, measuring location-sensitive DNS answers from content delivery networks [21] obviously requires multiple vantage points, whereas measuring how many domains use a certain DNSSEC configuration can be done from a single vantage point [3], [16], [17]. Also the scale of the measurement has an impact on the choice of the location and number of vantage points. Osterweil et al. [22], for example, follow the operational status of DNSSEC deployment since its rollout by means of distributed measurement points. Given the size of our daily dataset and the large amounts of queries we produce, we believe that unbridled duplication of our infrastructure would add an unwanted burden on the DNS system. We therefore foresee distribution as a future expansion aimed at studying specific aspects of the DNS behaviour.

VII. Data Sharing

We realise that the data we collect is highly valuable for other researchers. Also, it is clear that while Section IV-B illustrates that the impact our measurement has on the global DNS infrastructure is well within reasonable bounds, if lots of researchers were to set up similar infrastructures this would have a significant and possibly disruptive impact on the Internet. This means that we feel an obligation to make our data accessible to other researchers. We cannot make all our measurement data publicly available due to restrictions in the contracts under which we gain access to data for the TLDs currently measured. Nevertheless, we are working on two ways of making the data accessible:

1) We have set up a web portal[20] on which we will publish open access aggregate datasets. For example, all aggregate data sets for the case studies in this paper will be released through that portal. Examples of other aggregate data sets we intend to publish are daily counts of IP addresses in a TLD that geolocate to a certain country, counts of IP addresses that are inside a certain autonomous system (AS), the number of domains with at least one `AAAA` record (indicative of IPv6 use), etc.
2) We are in the process of setting up a program in which researchers can visit our group with the specific purpose of using the data we collect using the measurement infrastructure discussed in this paper. While the program is not ready yet, we already invite fellow researchers interested in using the data to contact us about visiting.

VIII. Conclusions and Future Work

Measuring the DNS is a potent tool for studying the day-to-day evolution of the Internet. For this reason, we set ourselves the task of actively collecting a long-term, large-scale data set that covers the main top-level domains on the Internet (including `.com`, `.net` and `.org`). When we started out, we had many questions about the feasibility of such a measurement. It was uncertain whether a sufficiently scalable infrastructure could be designed and implemented. Furthermore, if such a measurement were possible, how would it impact the global DNS infrastructure?

In this paper, we discussed the challenges of performing such a measurement and the choices we made while designing and implementing a novel active measurement infrastructure for this purpose. We have shown that our design scales to reliably measure even the largest top-level domain (`.com` at 123M names). Careful analysis of the traffic generated by the measurement system shows that while it generates a significant amount of traffic, the load on the global DNS infrastructure is at an acceptable level. Measurements started in February 2015, and collect daily data for all domains in `.com`, `.net` and `.org` (around 50% of all names on the Internet). Since then, the system has collected over 511 billion data points, totalling over 74TB of uncompressed data (10.1TB compressed).

To validate our measurement system and to illustrate the value of the data it collects, two case studies on the use of cloud e-mail services were performed. These studies show that a significant number of domains now use cloud mail services offered by Google, Microsoft and Yahoo. While – as expected – Google serves the largest number of domains, surprisingly, the use of Microsoft's Office 365 grows much faster. An investigation of the use of so-called SPF records for combating e-mail forgery also yielded interesting results. While both Google and Microsoft have detailed instructions on how to configure SPF when using their cloud services, use of SPF lags for Google users (at only 34.4%) compared to Microsoft users (over 92.4%).

As the case studies show, the data we collect can provide valuable insight in developments on the Internet, such as the use of cloud services. However, traffic analysis has shown that while the impact of our measurement on the global DNS infrastructure remains within reasonable bounds, it would be inadvisable for large numbers of network researchers to run a similar measurement. For this reason we are establishing a programme for visiting researchers to use the data we collect and will publish aggregate statistics on a dedicated web portal.

Future work – While our primary goal is to collect this data for research purposes, we realise that it has other applications, for instance in the security space. E.g., tracking over time what IP address mapped to which names can be a valuable tool in forensic investigations. While passive DNS is often used for this, we believe it is worthwhile examining if the data we collect can somehow provide a complementary view on this. We plan to work with Computer Security Incident Response Teams (CSIRTs) to explore this further.

Of course, we also strive to expand the coverage of our measurements by including additional TLDs. In some cases, such as the new generic top-level domains, we can gain access to the DNS zone files for these TLDs through ICANN's Centralized Zone Data Service[21]. In other cases, we need to collaborate with the TLD registry operators. This is especially the case for country-code TLDs (ccTLDs). We hope to convince these operators that collaboration is worthwhile by presenting our measurement infrastructure to them and demonstrating the value of the data both to them as well as to the wider Internet research community, by means of case studies.

[20] http://www.openintel.nl/

[21] https://czds.icann.org/en

Finally, we note that our operational experience shows that measuring data for domains that are remote to our measurement point (e.g. domains registered from China as mentioned in Section IV-A) has a performance impact. The distributed design of the measurement system allows for placing worker nodes in different locations. We intend to study the potential performance benefit this will give in the near future.

ACKNOWLEDGMENTS

The authors would like to thank Xander Jansen of SURFnet for his help in analysing the network flow data for our measurement infrastructure.

This work was supported by the EU-FP7 FLAMINGO Network of Excellence Project (318488) and by SURF, the Netherlands collaborative organisation for ICT in higher education and research institutes. The research leading to these results was made possible by OpenINTEL[20], a joint project of SURFnet, the University of Twente and SIDN.

REFERENCES

[1] S. Kitterman, "RFC 7208 - Sender Policy Framework (SPF) for Authorising Use of Domains in Email, Version 1," 2014.
[2] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications," *Proc. of the 22nd USENIX Security Symposium*, no. August, pp. 605–619, 2013.
[3] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and Its Potential for DDoS Attacks: A Comprehensive Measurement Study," in *Proc. of ACM IMC 2014*, 2014, pp. 449–460.
[4] T. White, *Hadoop - The Definitive Guide*, 4th ed. O'Reilly, 2015.
[5] Y. Yu, D. Wessels, M. Larson, and L. Zhang, "Authority server selection in DNS caching resolvers," *ACM CCR*, vol. 42, no. 2, p. 80, 2012.
[6] E. Osterweil, D. McPherson, S. DiBenedetto, C. Papadopoulos, and D. Massey, "Behavior of DNS Top Talkers, a .com/.net View," in *Passive and Active Measurement*. Springer, 2012, pp. 211–220.
[7] Internet Systems Consortium, "Internet Domain Survey," July 2015.
[8] V. Pappas, D. Wessels, D. Massey, S. Lu, A. Terzis, and L. Zhang, "Impact of configuration errors on DNS robustness," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 3, pp. 275–290, 2009.
[9] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains," *ACM Transactions on Information System Security*, vol. 16, no. 4, pp. 14:1–14:28, Apr. 2014.
[10] 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*, vol. 9, no. 5, pp. 714–726, Sept 2012.
[11] F. Weimer, "Passive DNS Replication," in *Proc. of the 17th FIRST Conference (FIRST 2005)*, 2005.
[12] C. Hesselman, J. Jansen, M. Wullink, K. Vink, and M. Simon, "A privacy framework for 'DNS big data' applications," November 2014.
[13] L. Deri, L. L. Trombacchi, M. Martinelli, and D. Vannozzi, "Towards a Passive DNS Monitoring System," in *Proc. of the 27th Annual ACM Symposium on Applied Computing*. ACM, 2012, pp. 629–630.
[14] I. N. Bermudez, M. Mellia, M. M. Munafo, R. Keralapura, and A. Nucci, "DNS to the Rescue: Discerning Content and Services in a Tangled Web," in *Proc. of ACM IMC 2012*. ACM, 2012, pp. 413–426.
[15] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman, "On Measuring the Client-side DNS Infrastructure," in *Proc. of ACM IMC 2013*, 2013, pp. 77–90.
[16] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "Making the Case for Elliptic Curves in DNSSEC," *ACM CCR*, vol. 45, no. 5, 2015.
[17] L. Zhu, D. Wessels, A. Mankin, and J. Heidemann, "Measuring DANE TLSA Deployment," in *Proc. of the 7th International Workshop on Traffic Monitoring and Analysis (TMA 2015)*, 2015, pp. 219–232.
[18] G. van den Broek, R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC Meets Real World: Dealing with Unreachability Caused by Fragmentation," *Communications Magazine, IEEE*, vol. 52, no. 4, pp. 154–160, April 2014.
[19] M. Lentz, D. Levin, J. Castonguay, N. Spring, and B. Bhattacharjee, "D-mystifying the D-root Address Change," in *Proc. of ACM IMC 2013*, 2013, pp. 57–62.
[20] S. Castro, D. Wessels, M. Fomenkov, and K. Claffy, "A Day at the Root of the Internet," *ACM CCR*, vol. 38, no. 5, pp. 41–46, Oct 2008.
[21] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig, "Web Content Cartography," in *Proc. of ACM IMC 2011*, 2011, pp. 585–600.
[22] E. Osterweil, M. Ryan, D. Massey, and L. Zhang, "Quantifying the Operational Status of the DNSSEC Deployment," in *Proc. of ACM IMC 2008*, 2008, pp. 231–242.

Roland van Rijswijk-Deij is a Ph.D. student at the University of Twente, the Netherlands, in the Design and Analysis of Communication Systems Group. He received an M.Sc. degree in Computer Science from the University of Twente in 2001. Roland also works for SURFnet bv, the National Research and Education Network in the Netherlands. His research interests include network security and network measurements, with a particular interest in DNS and DNSSEC.

Mattijs Jonker received a B.Sc. and M.Sc. in Computer Science from the University of Twente, the Netherlands. He is currently working toward the Ph.D. degree at the Centre for Telematics and Information Technology, University of Twente, the Netherlands, on the mitigation of DDoS attacks. His main research interests include network security, Internet measurements, and Big Data analytics.

Anna Sperotto is assistant professor at the Design and Analysis of Communication Systems Group of the University of Twente, the Netherlands. She received a Ph.D. degree from the University of Twente, in 2010, with the thesis titled "Flow-based intrusion detection". Her research interests include network security, network measurements and traffic monitoring and modeling.

Aiko Pras is professor in the area of Network Operations and Management at the University of Twente, the Netherlands in the Design and Analysis of Communication Systems Group. His research interests include network management, monitoring, measurements and security. He chairs the IFIP Technical Committee on Communications Systems, and is Coordinator of the European Network of Excellence on Management of the Future Internet (FLAMINGO). He is steering committee member of several conferences, including IM/NOMS and CNSM, and series/associate editor of ComMag, IJNM and TNSM.