Resilience of the Domain Name System

Computer Networks 139 (2018) 136–150

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Resilience of the Domain Name System: A case study of the .nl-domain

Lars Kröhnke[a], Jelte Jansen[b], Harald Vranken[c,d,*]

[a] Radboud University, P.O. Box 9102, Nijmegen 6500HC, The Netherlands
[b] SIDN Labs, P.O. Box 5022, Arnhem 6802EA, The Netherlands
[c] Open University of The Netherlands, P.O. Box 2960, Heerlen 6401DL, The Netherlands
[d] Radboud University, P.O. Box 9102, Nijmegen 6500HC, The Netherlands

ARTICLE INFO

Article history:
Received 24 July 2017
Revised 6 March 2018
Accepted 26 April 2018
Available online 27 April 2018

Keywords:
Domain Name System
Computer network reliability
Reachability analysis

ABSTRACT

In this paper we analyse the resilience of the Domain Name System (DNS). We study the impact on the availability of DNS data of certain domains for the users of the Internet when parts of the DNS infrastructure become unavailable. We perform our analysis on the level of autonomous systems and hence we explore the impact when an autonomous system (and all the routers, name servers and resolvers located in it) fails or when interconnections between autonomous systems fail. We provide a generic method to carry out this resilience analysis, in which we first identify the domain names within the analysed domain, the autonomous systems where the name servers and resolvers for this domain reside, and the interconnections and relations between these autonomous systems. Next, we simulate failure scenarios to analyse the impact on the reachability of autonomous systems and the corresponding DNS data when autonomous systems or connections between them become unavailable. Our method can identify bottlenecks and single points of failure that should be mitigated in order to improve resilience. We demonstrate our method in a case study for the .nl-domain and its underlying second-level domains.

1. Introduction

The Domain Name System (DNS) is used to translate domain names into IP addresses and as such it is a crucial part of the infrastructure of the Internet [1]. Without the DNS, many applications and services will not be able to map domain names on IP addresses and consequently they will not work as intended. The specifications of the DNS therefore require that DNS data is stored in a distributed and redundant way, preferably on servers located in different networks connected to the Internet, to improve the availability of DNS data in case of failure of some parts of the network [2]. However, there are many physical and logical locations on the Internet where something can break, either intentionally or unintentionally, resulting in DNS data becoming unavailable. Key questions addressed in this paper are to what extent the availability of DNS data is guaranteed when parts of the network malfunction, and to what extent the redundant storage of DNS data has actually been implemented.

The Internet is not a single network, but rather a network of networks [1,3]. Each of these networks forms an autonomous system (AS) and together these ASs form the Internet [4]. In July 2016 there were about 54,700 ASs [5]. Each AS roughly translates to one Internet Service Provider (ISP), however, there are exceptions as large ISPs may have a larger number of ASs [6] and also other organisations might own an AS. An AS is identified by a 32-bit AS number (ASN), which is managed and distributed by Regional Internet Registries (RIRs) who obtain these numbers from the Internet Assigned Numbers Authority (IANA) [7].

The DNS data is distributed over a large number of name servers which are located in various ASs. At any point in time an AS serving DNS data or connections between ASs may be interrupted for various reasons. For instance an AS may be disconnected from the Internet if the company owning the AS becomes insolvent. A connection may fail due to physical damage of the cable connecting two routers or failure of an upstream provider. Not only name servers may fail when a certain AS or connection becomes unavailable, but also public resolvers may become unreachable during such an incident, which would cause the whole DNS to be unavailable for users rather than only certain domain names.

Although failure of huge parts of the DNS sounds to be a worst-case scenario, multiple incidents have shown recently that the DNS

* Corresponding author at: Open University of The Netherlands, P.O. Box 2960, Heerlen 6401DL, The Netherlands.
E-mail addresses: i.bade@gmx.net (L. Kröhnke), jelte.jansen@sidn.nl (J. Jansen), harald.vranken@ou.nl (H. Vranken).

https://doi.org/10.1016/j.comnet.2018.04.015
1389-1286/

is vulnerable and that such failures do happen. The causes of these failures ranged from power outage and DDoS attacks to administrative errors. For instance, in October 2016 a massive DDoS attack was launched from the Mirai Internet-of-Things botnet against DNS name servers managed by Dyn, which consequently made numerous popular websites unavailable [8]. Also many anomalies have been identified that threaten the stability and reliability of the Border Gateway Protocol (BGP), the Internet's default inter-domain routing protocol that manages connectivity among ASs [9]. For instance, an administrative error occurred in February 2008 in Pakistan, when routers of Pakistan Telecom were manually misconfigured to announce a prefix which actually belonged to the IP address range of YouTube. This erroneous route was propagated to the rest of the Internet and subsequently all YouTube traffic was redirected to Pakistan. This caused that the AS of YouTube was unreachable for an estimated two-thirds of the Internet for about two hours [10,11]. It is well plausible that similar incidents can occur in the future.

In this paper we analyse the resilience of the DNS. We study the impact on the availability of DNS data of a certain domain for the users of the Internet when parts of the DNS infrastructure become unavailable. Our contributions are twofold: (i) we provide a method to carry out this resilience analysis, and (ii) we demonstrate our method in a case study for the .nl-domain and its underlying second-level domains. Our method can be applied to analyse whether a domain suffers from bottlenecks or single points of failure, that should be mitigated in order to improve resilience of this domain. We first identify the domain names within the analysed domain, the ASs where the name servers and resolvers for this domain reside, and the interconnections and relations between these ASs. Next, using a graph model of the identified ASs, we simulate failure scenarios to analyse the impact on the reachability of ASs and the corresponding DNS data from the viewpoint of the most used resolvers when ASs or connections between ASs become unavailable.

We perform our analysis on the level of ASs, which implies that we explore the impact when an AS (and hence all the routers, name servers and resolvers located in this AS) fail or interconnections between ASs fail. Although this approach is somewhat coarse grained, more detailed analysis at the level of individual routers, name servers or resolvers would be largely infeasible due to their sheer numbers and missing data. We therefore apply a two-step approach. Whenever the analysis at the AS level reveals issues, this pinpoints to directions where a more detailed analysis can be performed as a second step in which only a limited number of ASs and connections is involved. In this paper, we did not consider such detailed analysis yet. In fact, it turned out that in our case study for the .nl-domain, no major issues were identified that would require a more detailed analysis.

This paper is organized as follows. In Section 2 we briefly provide background information on routing and resolving of domain names to make the paper self-contained. In Section 3 we review related work. In Section 4 we outline our method for DNS resilience analysis, and in Section 5 we present a case study in which we apply the method to the .nl-domain. In Section 6 we discuss limitations of our method. In Section 7 we conclude the paper.

2. Routing and domain name resolving

This section provides background information, explaining basic concepts of routing and domain name resolving, covering DNS and BGP, with details on Internet exchanges and anycasting.

2.1. Domain Name System

In order to send a packet to a remote host over the Internet, the IP address of that destination host must be known. Since IP addresses are difficult to remember and interpret by human beings, domain names were introduced. The DNS translates domain names into the corresponding IP addresses. The DNS is described in many different RFCs which partly obsolete each other. The most important ones are RFC 1034 [2] and RFC 1035 [12]. The term DNS is used for both the Domain Name System itself and the protocol used in the system.

The DNS is a distributed system in which name servers and resolvers are the main components [2]. The name servers form a distributed database that stores information about domain names, such as IP addresses of hosts in that domain. Resolvers are systems capable of querying the database in order to resolve a domain name into an IP address. The name servers and the resolvers are located in different ASs all over the world.

The distributed DNS database has a hierarchical tree structure. Every name server is serving its own zone within the domain namespace. The basis of the domain namespace is formed by the root zone, denoted by a single dot (.). A top level domain (TLD) is a direct child of the root zone. In the beginnings of DNS, two types of TLDs were introduced: ccTLDs based on country codes from ISO 3166 [13], e.g. .nl for the Netherlands and .de for Germany, and gTLDs based on more generic terms, e.g. .com for commercial applications and .edu for educational institutes [14]. The direct child domains of a TLD are called second-level domains. Any child domain is a subdomain of its parent, e.g. example.nl is a subdomain of the .nl-domain, even though the term subdomain is commonly only used for domains in the third or lower level of the DNS tree. Every subdomain is delegated to a set of authoritative name servers serving the DNS data of that domain. To mitigate availability issues with the DNS, every zone should have at least two distinct authoritative name servers, preferably located in different networks [14].

When an end-user tries to resolve a particular domain name, e.g. example.nl, a stub resolver that is built into the application or the operating system of the user system, asks a recursive resolver (e.g. operated by an ISP) to perform a look-up. The recursive resolver sends a query to one of the root name servers, asking for the authoritative name servers of the TLD. The IP addresses of these root name servers are hard-coded in the resolving software. The root name server will respond with a referral to the name servers which serve the .nl-zone. The recursive resolver will next query one of those name servers in order to retrieve the IP address of example.nl, thereby traversing the DNS hierarchy tree up to the domain name in question. The resolution process can be optimised by using caching. A resolver caches responses obtained from name servers, such that these answers can be reused to answer subsequent queries. It may happen that the name servers for a particular zone are located in the very same zone, e.g. the authoritative name servers of example.nl might be ns1.example.nl and ns2.example.nl. In these cases a resolver is not able to resolve the IP addresses of the name servers, as it needs those IP addresses to be able to query the name servers for their own address. To circumvent this issue, glue records are being used, that are contained in the DNS response and specify the IP addresses of the name servers in question [2].

2.2. Border Gateway Protocol

The Border Gateway Protocol (BGP) is used by routers (gateways) at the border of an AS to share routing information with routers at the borders of neighbouring ASs [15]. Each AS owns a certain range of IP addresses, which is denoted by prefixes accord-

ing to the concept of classless inter-domain routing (CIDR) [16]. A prefix consists of an IP address and a mask length [17] indicating the number of fixed bits in the address. For example the prefix `192.0.2.0/24` is used for the range of IP addresses from `192.0.2.0` up to `192.0.2.255`. For simplicity and without loss of generality, in this paper we only refer to IPv4 addresses and no IPv6 addresses [18].

The basic working of BGP is that each BGP-speaking router at the border of an AS announces the prefixes of the AS to its direct neighbouring routers at neighbouring ASs in the network topology, which store this announcement together with the incoming port in their own routing table. These routers subsequently propagate the announcement to their neighbouring BGP-speaking routers, that in turn might propagate the announcement further to their neighbours, according to their configuration and policies. This way a global routing table is created where every BGP router in theory knows at least one path to every other AS in the network topology. The propagation of announcements is based on trust, which does not seem to be a big issue in practice as fraud, such as announcing a prefix which is not owned by an AS, is easily detected and traceable [19–21].

When transmitting a packet, a router looks up the best route to the packet's destination in its routing table and forwards the packet to the next router on this path. Every router has its own distinct routing table [1], dependent on the location of the router within the Internet. This routing data can be used to retrieve an overview of the connectivity of the Internet [1]. Ideally, this would allow to obtain an image of the Internet topology. However in practice not all routers have the same image of the Internet at the same point in time, as propagation of announcements with BGP needs some time to converge.

Each AS may have its own routing policies and preferences, e.g. preferring one route over another. These policies mainly rely on commercial contracts between the owners of different ASs [22]. Such contractual agreements are also used to classify the connections between ASs. Essentially, this classification boils down to who is paying whom for providing the connection. The commonly used classification scheme is based on customer-provider, peering, and sibling relationships [22,23]. In customer-provider relationships the customer is paying the provider for providing upstream connectivity. Also transit traffic transmitted via a provider AS has to be paid for. A provider transits traffic for its customers, but a customer does not transit traffic between two of its providers. In peering relationships a pair of peers agree to exchange traffic between their respective customers free of charge. However, peers do not transit traffic for each other. Peering relationships are mostly found between ASs of roughly the same size as otherwise one AS would send considerably more traffic to the other one. However, there are some exceptions where one peer pays another peer [24–27]. In a sibling relationship the ASs have a mutual-transit agreement and they provide connectivity to the rest of the Internet for each other. The sibling relationship applies to ASs with common administrative boundaries, e.g. ASs belonging to the same organisation, or to small ISPs who are located close to each other and who cannot afford additional Internet services for better connectivity. A sibling relationship may also provide a mutual-backup agreement where the ASs provide backup connectivity to the Internet for each other in the event that the connection of one AS to its provider fails.

Fig. 1 shows an example AS topology, using a graph representation where nodes correspond to ASs and edges indicate traffic flow. The form of an edge indicates the relationship between the connected ASs. Directed edges do not indicate a direction of the data flow, but the direction of the money flow from the source AS to the destination AS. In the figure, A is provider for B and C; B is provider for D and E; C is provider for E and F; B and C peer

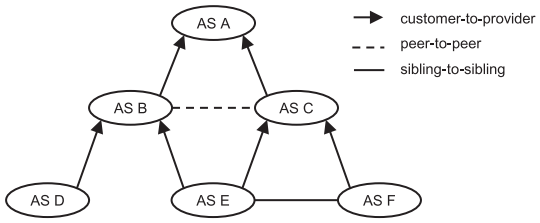

Fig. 1. Relationships between ASs.

with each other; and, E and F are siblings. Paths between two ASs are only valid if every transit provider in the path is paid by one of its direct neighbours on the path. For instance, the path B-E-C is invalid, since E would have to provide transit traffic without being paid for. Also, the path D-B-C-A is invalid, since C is not being paid for providing transit traffic, while the path D-B-C-F is valid. If B and C would have a sibling relationship instead of a peering relationship, then the path D-B-C-A would be valid.

ASs that have connections to multiple transit providers are called multihomed, whereas ASs with just one transit provider are called singlehomed. Using multiple transit providers makes sense as this redundancy ensures that the AS remains connected in case of failure of one of the transit ASs. Routing policies may be implemented to use one preferred transit provider, while the other transit providers act as backup in case the preferred transit provider fails. The notion of singlehomed and multihomed AS is however a bit misleading, as every AS has to have at least two upstream providers in order to obtain an ASN by IANA [28], however in practice a single upstream provider can be used.

Due to the interconnections between ASs and the routes being advertised, it often happens that route loops are created. To manage this problem, routes contain the path of ASs that is traversed to reach a certain destination prefix. If a router receives a route with its own ASN in the path, it can simply ignore this route.

2.3. Internet exchanges

The European Internet Exchange Association defines an Internet Exchange Point (IXP) as a network facility that enables the interconnection and exchange of Internet traffic between more than two independent ASs [29]. Internet Exchanges are thus central places to facilitate ISPs to easily peer with each other. The worldwide biggest IXPs, such as the AMS-IX in Amsterdam, DE-CIX in Frankfurt and LINX in London, can in terms of handled traffic even be compared to the largest Tier-1 providers in the world [3,30]. An IXP can be considered as a giant switch where every participant connects its own router [31]. That way, the routers of the participants are directly connected to each other and can directly exchange routing information via BGP sessions, which is referred to as bilateral peering. Bilateral peering however requires a separate BGP session for every peering connection. This can be circumvented by using route servers that allow multilateral peering, where participants establish a single BGP session with a route server, which then broadcasts and sometimes filters according to the operators policies all incoming routes to the connected routers. Route servers are however not meant to forward any traffic [3,32] and are therefore also called route reflectors. These route servers, or rather the ASs the route servers belong to, do not appear in the path of certain routes, which makes it complicated to retrieve information about the way routers are connected to each other.

2.4. Anycasting

Anycasting means that multiple routers at different locations within the global network topology announce the same IP address prefixes to their neighbouring routers. As a result routers within different parts of the Internet will route the traffic belonging to a certain IP address to different networks. This shortens the mean path length to reach a certain IP address [33].

A disadvantage of this technique is that the servers whose routers announce the same IP address have to stay synchronised, because otherwise the responses of requests to the same IP address may be different according to the network location of a client. Furthermore it should only be used with stateless protocols as changes in the network topology might cause the best reachable server to change during a session. That means, that the first packets of an established session reach a different side than the last packets of the same session, causing the session to be destroyed. However, the advantages of this technique are worth the effort. Next to faster response times due to shorter network paths, anycasting can also be used for efficient load balancing [34] and the introduced redundancy ensures availability of the served data, even when one of the anycast nodes stops working. Anycasting is therefore also used in the DNS infrastructure [35–37].

3. Related work

A lot of previous research has focused on obtaining the network topology of the Internet at the AS level. This includes research by e.g. Gao [22] and Magoni and Pansiot [38]. In this previous research different data sources have been used for inferring the AS-level topology of the Internet, but it is not known yet which of these data sources produce the best image of the actual topology [39]. Although different data sources have been utilized, they are all dependent on data collected from the BGP which was never intended to reveal the network topology and should therefore be used with care [28]. Probably one of the most complete views on the connections of ASs is given by the methodology used by Zhang et al. [40]. Unfortunately, these researchers discontinued their topology analysis in the beginning of 2015. Outdated data sets dating from September 1999 up to February 2015 are however still available at [41].

To the best of our knowledge, no data exists which perfectly shows the connections of ASs within the Internet and up to now there is no way known to perfectly infer this topology [42]. This is due to the way the Internet is constructed. There is no authority which manages the Internet at the topmost level. Some researchers even stated that it is impossible to obtain this topology [42]. This may be the reason that there has been relatively little research on this in recent years. More recently, far less research directly aiming at inferring the network topology has been performed, although this has been addressed in studies with different goals, such as finding critical regions and paths in a network [43]. A good overview of research related to the inference of the network topology of the Internet at the AS level is given in a survey by Haddadi et al. [44]. Another, more recent survey by Motamedi et al. [4] inventories topology inference techniques at various levels.

Research on the resilience of the Internet, e.g. the work performed by Rexford et al. [17], focuses mainly on the availability of the most popular domain names, based on the traffic received at the prefixes corresponding to those domains, rather than on the availability of the domain names within the DNS. This is however also a very important factor, as without the DNS entries also the servers pointed to by that DNS entry are not reachable, except for those users who by chance know the IP addresses belonging to that domain.

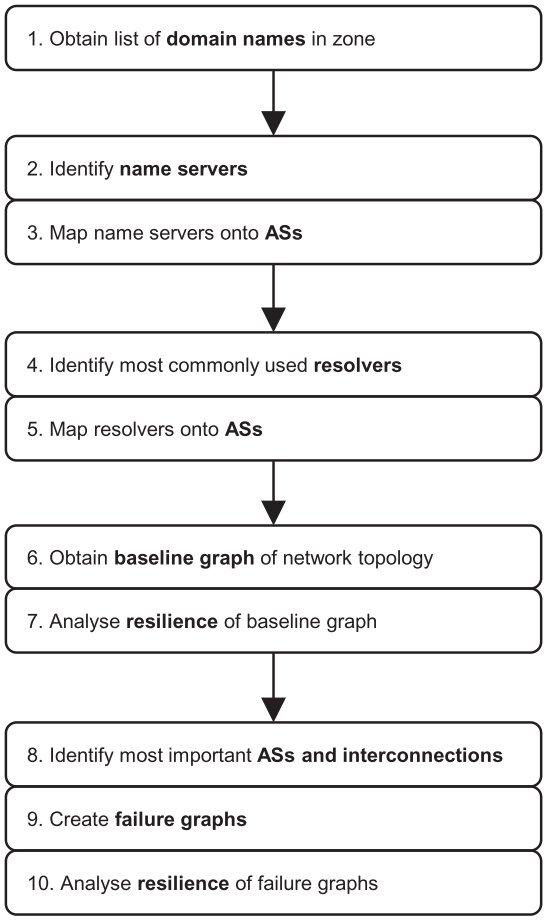

Fig. 2. Our analysis method.

Research on the resilience of the DNS has focused mainly on the performance of name servers serving the root zone [45] and not on the impact of certain failure scenarios on a TLD or even second-level domains. Other recent research on the availability of DNS investigates DDoS attacks on the DNS [46–48].

4. Method for DNS resilience analysis

In this section we present our method for DNS resilience analysis. Our method includes 10 steps, as outlined in Fig. 2 and detailed in the following subsections. We wrote Python scripts for all steps (except step 1), and hence the method can be fully automated [49]. Since some steps are compute intensive and since the data within the DNS and BGP routing tables are constantly changing, the method cannot be performed in real time. Therefore a copy of the required DNS and BGP data should be made at a certain point in time, such that the method can work afterwards for as long as it takes with consistent data. Also, storing copies of the data allows to rerun the analysis at later times or to analyse differences in the data at different points in time.

4.1. Obtain list of domain names in zone

The first step is to obtain a list of all domain names in the zone that is subject of analysis. Although our analysis method is generic, it is executed primarily by the registry that manages the zone file. In our case, our research was initiated by and performed at SIDN, registry of the .nl TLD, and hence we had a copy of the zonefile of the .nl-zone.

All DNS data published in the global DNS namespace is in principle publicly accessible by querying the DNS. However, the DNS is not a searchable directory, but works more akin to a key/value store. Hence, it is generally not possible to retrieve the full contents of a zone through the DNS itself. Different operators have different policies on publishing such data. For a number of TLDs, this data is accessible through an AXFR DNS request, which initiates a full zone transfer. Other TLDs offer a separate download page, with usage restrictions that the downloader needs to agree to. Yet other TLDs keep their full zone data secret, for security or privacy reasons.

4.2. Identify name servers

The zonefile contains all registered domain names along with their name servers and possibly glue records. The name servers in the zonefile however may not match the name servers actually in place. Therefore, we use the zonefile only to extract all currently registered domain names in the zone. The name servers of a domain name and corresponding IP addresses of these name servers can be retrieved by actively resolving the domain names of the name servers. This can for instance be achieved with the commandline tool `dig`. A more efficient way is to use a tool such as `spark` [50] (developed by SIDN) that can resolve many DNS queries in parallel.

As the metric for importance of a name server we use the number of domain names hosted on that name server. Other metrics could be to identify the most important domain names and the name servers that serve these. Some domain names may have been acquired by squatters, and hence are used rarely and could be ignored in the analysis. However, in general it is not evident what criteria should be applied to rank the importance of domain names. Data sources such as the Alexa-top 500 websites, or data from search engines, social networks, advertisements, or DNS (e.g. [51]) may be considered to identify the most frequently visited domain names. An even better alternative may be to rank according to the impact of domain names instead of popularity. For instance, the availability of domain names of banks or public services is more important for private users than the domain names of their private home pages, while the availability of domain names of web stores and companies has direct financial impact on their businesses. To the best of our knowledge, such rankings however are not publicly available. We therefore only use the amount of domain names hosted as the metric for importance of a name server, and not the popularity or impact of domain names.

4.3. Map name servers onto ASs

We next map the IP addresses of the name servers onto the ASs these IP addresses belong to. For each name server, we obtain the corresponding ASN. To better understand and be able to interpret the results of our analysis, we also investigated the companies these ASNs belong to. For this purpose, the AS-to-organization mapping provided by CAIDA [52] can be used. This dataset is based on WHOIS data associated with an ASN [53].

4.4. Identify resolvers

Besides the name servers, also the (recursive) resolvers used to query the name servers are important, since our goal is to analyse the reachability of authoritative name servers in the case of an incident. These name servers must be reachable from recursive resolvers. In turn, of course, the recursive resolvers must be reachable for their users (i.e. stub resolvers). It is hardly possible to obtain information about stub resolvers as this information is only available at individual user systems and ISPs. However, the information on users and stub resolvers is aggregated into the information of the recursive resolvers, and hence our level of analysis allows to abstract from the details at user level.

Information on recursive resolvers also is not publicly available. The registry of the zone can see all DNS queries received at its name servers, which can be analysed to identify the resolvers that issued the queries. As mentioned, our research was performed with SIDN, which operates the ENTRADA database [54] in which all DNS queries received at the authoritative name servers of the .nl-zone are stored.

Also at the level of recursive resolvers it is nearly impossible to analyse reachability of the authoritative name servers for all resolvers due to the huge amount of distinct resolvers being used. We therefore focus on the most important ones, i.e. the resolvers that issue the largest number of queries for a zone. Defining the importance of a resolver appropriately is a challenge of its own. A rather naive approach is to just consider those resolvers that issue the most requests to the authoritative name servers of the zone. However, we observed that some resolvers issue large numbers of requests automatically for various reasons. Some resolvers are constantly querying whole zones in a linear fashion, either to generate profit (e.g. domainers, who register potentially valuable domain names to sell them later) or for research purposes [55,56]. Other resolvers send the same query multiple times in very short intervals, which may be due to misconfiguration or absence of caching at the resolver, or resolvers send the same query to multiple authoritative name servers and just use the fastest reply for optimizing the performance of the resolver at cost of the performance of the authoritative name servers. Also, resolvers which validate DNSSEC entries send additional queries to name servers in order to obtain the necessary records for validation of the resource records.

Hence, the challenge of selecting the most important resolvers is to identify those resolvers that are configured correctly and whose queries originate from human users rather than automated applications. We therefore first identify those resolvers that send basic query types (A, AAAA, CNAME, MX, NS, PTR, SOA, and TXT), thereby excluding DNSSEC related requests and other unusual query types. We remove duplicate queries, i.e. identical queries sent in short time intervals (e.g. less than 5 min) from the same resolver. Next, we exclude resolvers that issue only queries of type NS as these most probably belong to domainers. This behaviour would also be seen from resolvers that implement qname minimisation [57], but at the time of this research, deployment of that technique was negligible. We consider that resolvers are shared by many users and the normal behaviour of resolvers therefore is that more popular domain names are queried more often while less popular domain names are queried only a few times a day. We therefore exclude resolvers that act as crawlers (i.e. resolvers that issue a high percentage of queries for distinct domain names), resolvers that act as monitors (i.e. resolvers that often query only a small amount of domain names), and resolvers that act as scanners (i.e. resolvers that scan a zone in a linear fashion). We also exclude resolvers that cause mainly NXDomain (non-existent domain) responses, since these are probably used by botnets that were applying domain name generation algorithms to generate pseudo-random domain names of which many are non-existing. The downside of excluding these resolvers is that also the legitimate queries originating from those resolvers are not taken into account. However, the number of legitimate queries that are dropped this way is negligible within the greater set of legitimate queries. Nevertheless, a better approach is to filter out the legitimate queries, which we consider as a topic for further research.

4.5. Map resolvers onto ASs

We map the IP addresses of the selected resolvers onto the ASs these IP addresses belong to, in the same way as we mapped the IP addresses of the selected name servers onto ASs.

4.6. Obtain network topology of ASs and relationships

The most difficult task is to obtain an accurate network topology of ASs where the selected name servers and resolvers are located in and their relationships. There is no data available that perfectly represents the Internet network topology, which is mainly due to the fact that the Internet is a global self-managing network without a central authority that knows everything about the connected parties and their relationships to each other [42]. BGP by its nature always presents an incomplete topology, since a BGP-speaking router forwards only a single 'best' route for each prefix to its neighbours, according to the router's export policy that is often determined by business relationships and therefore kept private. Roughan et al. [28] give a detailed summary of the problems associated with inferring the AS topology by means of BGP traffic.

To the best of our knowledge, the most complete, publicly available view on the Internet network topology in terms of AS relationships is currently provided by CAIDA [23]. This CAIDA dataset provides connections inferred from several sources such as the RouteViews project [58], RIPE RIS [59] and CAIDA's ark monitors [60] (see [23,61–63] for more detailed information and further background about the methodology used to infer this dataset). A big advantage of the CAIDA dataset in contrast to other available datasets (such as the 'Neighbours'-tool of RIPEstat [64]) is that this dataset also provides the relationships between two interconnected ASs, enabling us to limit our analysis to the paths on which traffic actually might be exchanged, rather than analysing all existing connections.

However, the CAIDA dataset has some limitations too. Since the actual AS topology changes dynamically, it is difficult to combine data from different sources at different points in time. Although it is tried to synchronise the data as well as possible, some errors will be introduced. The CAIDA dataset does not specify the locations at which peering between two neighbouring ASs takes place, such as certain IXPs. We assume that all members of an IXP peer with each other, which is the most plausible scenario although there are some special situations where this simple rule may not apply. Also, the CAIDA dataset does not specify sibling relationships.

Further limitations are that we only analyse logical connections between ASs, rather than physical links. The latter would be more interesting as several logical connections may originate from the same physical link. Also it is possible that several physical cables are bundled in the same underground pipe, which might get damaged during construction works. However, this information is kept private by the owners of physical infrastructure. Due to the use of anycasting, logical connections may appear in the obtained topology that are physically distributed over two different networks. (The use of anycast is included implicitly in the AS network topology; an AS that is applying anycast will show up as being well connected.)

We represent the network topology in a graph model such as shown in Fig. 1, in which vertices represent ASs and edges represent connections between ASs. The relationships between ASs are represented by labels connected to the endpoints of each edge that indicate whether the AS acts as customer, provider, peer, or sibling on this connection. If there is evidence that peering takes place at a certain IXP, we add a second label to such edges that indicates the IXP.

4.7. Analyse DNS resilience

We analyse the reachability from the ASs that contain the identified most import resolvers to the ASs that contain the name servers. Let set R contain the ASs of the identified most import resolvers, and set S the ASs of the name servers. For each pair (r, s) with $r \in R$ and $s \in S$ we compute the shortest path in the baseline graph from r to s. The labels attached to the edges in the graph indicate whether a path is valid or invalid. As explained in Section 2.2, a valid path is a path that consist of zero or more customer-to-provider edges, followed by zero or more peer-to-peer edges, followed by zero or more provider-to-customer edges, where sibling-to-sibling edges may occur at any position. This complicates the analysis and this is not supported in standard libraries for graph analysis. We therefore implemented a dedicated library to perform the required analysis (see [49] for details).

Next to analysing whether there is a path between two ASs, we also analyse the length of the shortest path as a quality measure of the network. The path length does not impact the reachability, but does impact performance. This may reveal some ASs used for storing DNS data which are poorly connected to those ASs requesting this data. As another quality measure also the amount of different paths could be considered, however this would require to compute all possible paths which is unfeasible for large graphs.

4.8. Identify most important ASs and interconnections

In the process of resolving a DNS request, three different sources of possible network failures can be identified:

1. failure of the resolver;
2. failure of any part of the network necessary for transmitting the request to the authoritative name server, i.e. transit providers or connections;
3. failure of the authoritative name server.

Ideally one would want to know the impact of failure of any part of the infrastructure. Since our analysis is on the level of ASs, we consider that ASs containing resolvers, ASs providing transit, ASs containing authoritative name servers, and interconnections between ASs can fail. Despite this abstraction, the amount of possible failures still is huge. We therefore focus on those ASs that host name servers for the largest numbers of domain names, since these ASs presumably have the highest impact on the availability of DNS data. We also focus on those ASs and interconnections that are most commonly traversed in the baseline, since these are the most important transit providers and have the highest impact on the availability of paths between resolvers and name servers.

The criteria for selecting ASs and interconnections can be refined further (but are outside the scope of this paper). For instance, criteria can be considered in which geographical location plays a role. Co-location of name servers might be an interesting option, and one could analyse failure of a data center and the name servers located in that data center rather than all name servers located in an AS. Furthermore, as a data center might host servers located in various ASs, this would add another aspect to the analysis. For interconnections, actual physical connections such as submarine cables can be considered that cluster logical links between ASs, and one could analyse the impact of cutting such a cable which might affect numerous logic links.

4.9. Create failure graphs

We recognize that there is vast scientific literature on Failure Mode and Effects Analysis (FMEA). The main objective of FMEA is to identify potential failure modes of different components, evaluate the causes and effects, and determine what could eliminate or

reduce the chance of failure [65]. FMEA was first developed in the 1960s by the aerospace industry [66]. Since then, FMEA has been extensively used in a wide range of industries, including computer networks [67].

Instead of addressing FMEA by considering failure modes of components in the Internet, we take an aggregated approach in which we abstract from components. We consider failure scenarios in which ASs and interconnections fail. We consider scenarios where either one AS fails, one interconnection fails, or multiple interconnections fail that are all connected to an IXP. For each scenario, we make a copy of the baseline graph and remove the corresponding AS or interconnection(s).

We do not consider the failure of resolvers in our analysis, since all ASs and domain names would be unreachable for a user if the AS containing the resolver fails. It might happen that an AS serves both sides of a DNS request, when the resolver and the authoritative name server are located within the same AS. In these cases no conclusions can be drawn for this particular AS and therefore we do not consider such cases.

As mentioned before in Section 4.6, we apply the most complete, publicly available view on the Internet network topology in terms of AS relationships as provided by CAIDA, which however is inherently incomplete. Consequently, also our baseline graph and failure graphs of the AS network topology with the selected name servers, resolvers, and relationships, are inherently incomplete. In addition, BGP-speaking routers may forward back-up routes in case of failures in practice. These back-up routes only appear in case of failure, and are normally invisible and not part of the CAIDA data or any publicly available dataset.

4.10. Analyse DNS resilience with failures

For each failure graph, we analyse whether ASs containing name servers have become unreachable. We also recompute the shortest paths between the ASs containing the identified most import resolvers and the ASs containing the name servers. Changes in the shortest path length indicate that the performance of the network is affected. In order to reduce computations, we cache the shortest paths in the baseline graph. When recomputing the shortest paths in a failure graph, we first check whether the shortest paths in the baseline graph are affected in the failure graph. This optimisation drastically reduces the number of shortest paths that have to be recomputed, and hence the computation time.

5. Case study of .nl

We applied our method for DNS resilience analysis in a case study on the .nl-domain, the ccTLD of the Netherlands, and its second-level domains.

5.1. Obtain list of domain names in zone

SIDN provided a copy of the zonefile of the .nl-zone on June 2, 2016, which contained 5,626,381 registered domain names within the .nl-zone. We checked all these domain names by resolving them. Some domain names could not be resolved, because the authoritative name server just contained the domain name without further resource records or did not have any data at all regarding that domain name [68]. Most of these errors are due to domain names that have been registered but never used by their owners, and therefore they are also not well configured. In total 5,364,788 domain names could be resolved and we used these in our analysis. We did not distinguish domain names by their relative usage or worth; for the purposes of this study we considered each domain name equally important.

5.2. Identify name servers

Next to the registered domain names, the zonefile also contains the corresponding name servers and possibly glue records. We only used the domain names from the zonefile. We queried the DNS to obtain the name servers for all domain names in the zonefile, and subsequently to obtain the IPv4 addresses of these name servers. By using the parallel resolving provided by `spark` all required (roughly 6 million) DNS queries could be resolved within 1.5 h. This process could have taken several days without parallelizing.

In total we obtained the hostnames of 69,996 name servers. Again, it turned out that some hostnames could not be resolved, and we excluded these name servers, resulting in 68,060 resolvable domain names of name servers. Further investigation revealed that some of the domain names of name servers point to the very same IP address, such that in fact only 45,031 distinct name servers are responsible for serving the whole .nl-zone. Hence, a rather small amount (roughly 45 K) of name servers is utilized to serve the 5.6 million domain names within the .nl-zone.

For 16,394 domain names, which is 0.3% of the analysed domain names, the list of name servers specified in the .nl-zonefile did not exactly match the list of name servers obtained by querying the DNS. This may be due to updates that happened in the time frame between retrieval of the zonefile and querying the DNS, or by administrators updating their infrastructures but not their records at SIDN. These mismatches should be investigated further, but that is out of scope for this research.

On average every name server serves 311 domain names. This however is not a uniform distribution. Only a few name servers host most of the domain names. The name server that hosts the most domains, serves 437,228 domain names, which is about 8% of the whole zone. Only roughly 10% of the name servers host more than 100 domain names. The median is 5 domain names per name server, and almost 75% of the name servers host less than 10 domain names. Hence, most of the name servers just serve a small amount of domain names.

5.3. Map name servers onto ASs

We mapped the IP addresses of the name servers onto the corresponding ASs by using the GeoLite database provided by MaxMind [69]. The main advantage of this method, in contrast to other services providing the same information as the RIPE database [70] and the service provided by Team Cymru [71], is that the dataset can be downloaded and therefore the look-up of ASNs can be done in memory, rather than issuing queries to network services, which offers a major performance speed-up. As a backup method, the 'Network Info' functionality of the RIPE database was used. The ASN lookup failed for 31 name servers that were not included in the MaxMind and RIPE database.

The identified name servers are located in 3806 different ASs. Also in this case, there is no uniform distribution: only a small amount of the ASs (roughly 3%) host more than 100 name servers, while the majority of the ASs (roughly 85%) host less than 10 name servers. A similar effect can be observed in Fig. 3, which shows for each of the identified ASs the amount of domain names attributed to it. A domain name is attributed to an AS if a name servers that hosts the domain name, is located within that AS. About half of the identified ASs host less than 10 domain names.

A further interesting observation is that 1,715,627 domain names, which is 32% of the analysed domain names, have all their name servers in a single AS. The situation is even worse for 16,095 of these domain names (0.3% of the analysed domain names), which are served by a single name server and hence there is no backup in case this name server fails. This is undesirable and

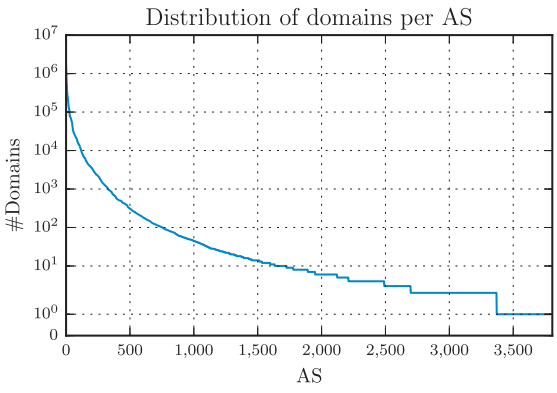

Fig. 3. Distribution of amount of domain names hosted on name servers located in distinct ASs.

against the requirements of SIDN, which state that for each domain there must be a primary name server and at least one secondary name server, and these name servers should be redundant machines on separate (sub)networks [72]. This situation could occur since only the domain names of name servers have to be reported to SIDN when registering a domain name, while these domain names still may be mapped to the same IP address.

For each ASN, we identified the companies these ASNs belong to using the AS-to-organization mapping dataset provided by CAIDA [52] with some manual corrections. As expected, the ASs hosting the most domain names belong to big Dutch hosting- and infrastructure providers, such as Schuberg Philis, TransIP, Hostnet, and LeaseWeb. We also looked at the geographical locations of the name servers using the GeoLite Country database by MaxMind Inc [69]. An interesting observation is, that although more than 75% of the domain names within the .nl-zone are hosted in the Netherlands, less than 50% of the name servers are geographically located there. This indicates that those name servers hosting a lot of the domain names are located within the Netherlands.

5.4. Identify resolvers

We identified the most important resolvers for the .nl-zone using ENTRADA [54], an open-source platform for storing and analysing large amounts of DNS traffic developed by SIDN Labs. When we performed the analysis in the first week of June 2016, ENTRADA had stored the DNS queries received at two out of the seven authoritative name servers of the .nl-zone for about 2 years. In order to keep the computation time required for data analysis within reasonable bounds, and to synchronise the different data sources used in our analysis, we only analysed the 3 billion DNS queries received in the period June 1–7, 2016.

We performed our analysis directly at the level of ASs. The main reason for doing so is that ENTRADA stores the ASN of the source IP address in each DNS query. Hence, we did not first have to identify the most important resolvers and next map these onto ASs, but we could select directly the ASs were most DNS queries originated from. ENTRADA provides a simple SQL interface to retrieve the stored data.

We first counted the amount of DNS queries issued by resolvers in different ASs, and we selected the top 30 ASs with the highest counts. The resolvers in these 30 ASs issued 65% of the 3 billion analysed DNS queries. The top 3 ASs are responsible for 26% of the DNS queries: AS15169 owned by Google (10%), AS49544 owned by i3d (10%), and AS20857 owned by Transip (6%).

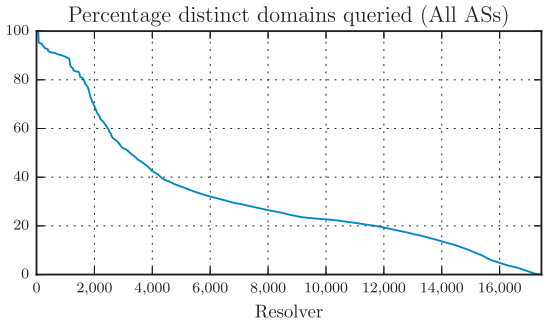

Fig. 4. Distinct domain names queried in first week of June 2016 by resolvers.

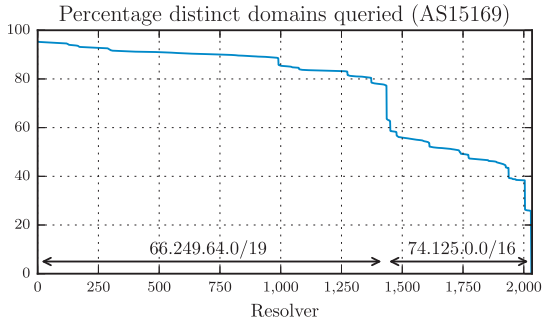

Fig. 5. Distinct domain names queried in first week of June 2016 by resolvers located in AS15169 operated by Google.

We next filtered the DNS data as explained in Section 4.4 to obtain only those DNS queries that originated from human users rather than automated applications.

Fig. 4 shows for each of the identified resolvers, the percentage of distinct domain names queried per resolver in the first week of June 2016. The figure clearly shows that there are different categories of resolvers. For instance, about 2000 resolvers issue queries for domain names of which more than 70% are distinct, while about half of the resolvers issue queries for domain names of which less than 30% are distinct. However, there are no strong boundaries between these categories. In our analysis we considered resolvers to be crawlers when they query more than 90% distinct domain names, and we considered resolvers as monitors when they query less than 1% distinct domain names. We manually identified resolvers as scanners when they generate disproportionate amounts of queries. In the filtering process, we ignored small resolvers that issue less than 10,000 queries to avoid that such resolvers would be considered as crawlers.

The filtering excluded 61% of the DNS queries, which illustrates that a large part of DNS queries originate from domainers, crawlers, monitors, and scanners. We again counted the amount of DNS queries issued by resolvers in different ASs, and selected the top 30 ASs with the highest counts as a second set. The resolvers in these 30 ASs issued 48% of the filtered DNS queries. The top 3 ASs are responsible for 21% of the DNS queries: AS15169 owned by Google (13%), AS8075 owned by Microsoft (5%), and AS32934 owned by Facebook (3%).

An illustrative example is AS15169 operated by Google. Fig. 5 shows the percentage of distinct domain names queried per resolver located in this AS. In this figure clearly three steps can be identified. The first group of resolvers issue queries of which almost 80% or more are distinct, that can be considered as crawlers. It turned out that these resolvers all have prefix `66.249.64.0/19` that is used by Googlebot [73]. The second group of resolvers issue queries of which roughly 40% to

Table 1
Selected resolver locations with reachability analysis in baseline topology.

Resolver location (ASN)	Owner	#Unreachable ASs	#Unreachable domains	Mean length of shortest path
AS1103	SURFnet, The Netherlands	2	0	1.182
AS2637	Georgia Institute of Technology	3	0	2.189
AS3215	Orange S.A.	6	1	2.836
AS3320	Deutsche Telekom AG	5	1	1.819
AS3356	Level 3 Communications, Inc.	3	0	1.741
AS4134	China Telecom Backbone	1	0	2.099
AS5432	Proximus NV	1	0	2.277
AS6830	Liberty Global Operations B.V.	4	1	1.909
AS8075	Microsoft Corporation	1	0	1.645
AS8737	KPN B.V.	1	0	2.886
AS8972	PlusServer AG	1	0	2.223
AS9121	Turk Telekomunikasyon Anonim Sirketi	1	0	2.097
AS9143	Ziggo B.V.	1	0	1.940
AS13127	Tele 2 Nederland B.V.	5	1	2.037
AS13238	YANDEX LLC	2	0	1.663
AS13414	Twitter Inc.	1	0	1.791
AS13335	CloudFlare, Inc.	1	0	1.645
AS14618	Amazon.com, Inc.	1	0	2.652
AS15169	Google Inc.	4	1	1.651
AS16276	OVH SAS	3	0	1.667
AS16509	Amazon.com, Inc.	1	0	1.825
AS17204	Nominum, Inc	1	0	2.359
AS20857	Transip B.V.	1	0	1.553
AS20940	Akamai International B.V.	1	0	1.621
AS23033	Wowrack.com	1	0	2.500
AS24793	NL Hosting Internet	1	0	2.223
AS24940	Hetzner Online GmbH	1	0	1.611
AS31615	T-mobile Netherlands bv.	1	0	2.001
AS32934	Facebook, Inc.	1	0	1.759
AS34173	SafeBrands S.A.S.	1	0	2.219
AS35470	XL Internet Services B.V.	1	0	2.639
AS36351	SoftLayer Technologies Inc.	1	0	1.495
AS36647	Yahoo	1	0	2.650
AS36692	OpenDNS, LLC	1	0	1.908
AS49544	i3d B.V.	1	0	1.711
AS55967	Beijing Baidu Netcom Science and Technology Co., Ltd.	1	0	2.254
AS60781	LeaseWeb Netherlands B.V.	2	0	1.825
AS197902	Hostnet B.V.	1	0	1.773
AS393406	Digital Ocean, Inc.	2	0	2.140
			μ_{path}	**2.000**

60% are distinct. These resolvers have prefix `74.125.0.0/16` which is used for Google's Public DNS Service. These resolvers all share the same cache, which causes that roughly half of the queries to authoritative name servers are distinct [74]. The third group of resolvers issue queries of which less than 25% are distinct. These resolvers can be considered as resolvers that resolve the queries issued by regular users. In the filtering process the first group of resolvers, that generate about 50% of the queries issued from AS15169, is filtered out.

The filtered and unfiltered list of top 30 ASs show a lot of overlap: 21 ASs occur in both lists. We combined both sets, resulting in a list of 39 most important ASs that we used in the subsequent steps. This list is shown in Table 1: the first column shows the ASN and the second column shows that organisation that operates the AS. The list contains ASs belonging to all kinds of different organisations working with the DNS, such as ISPs, hosting providers, owners of Internet infrastructure, search engines, and content providers. Hence, this list is representative for many use cases of the DNS.

5.5. Obtain network topology of ASs and relationships

We inferred the topology of ASs from the dataset provided by CAIDA. This topology contained 54,466 ASs with 488,140 connections.

5.6. Analyse DNS resilience

We analysed the DNS resilience in the baseline AS topology by analysing the reachability from the ASs containing the most important resolvers (set R) to the ASs containing the name servers (set S). Let set $S_r = \{s \in S | d(r,s) \neq \infty\}$ (where distance $d(r, s)$ is the length of the shortest path between r and s) be the set of ASs containing name servers that are reachable from r. Likewise, let set $U_r = \{s \in S | d(r,s) = \infty\}$ be the set of ASs containing name servers that are unreachable from r. For each $r \in R$ we computed three aspects:

- the amount of unreachable ASs, i.e. $|U_r|$
- the amount of unreachable domains, i.e. $|D(U_r) \backslash D(S_r)|$, where $D(S)$ is the set of domain names that are hosted on name servers located in S
- the mean of the shortest path lengths from r to every $s \in S_r$, where the path lengths are weighted by the number of domain names served by name servers located in s, i.e. $(\sum_{s \in S_r} |D(s)| \cdot d(r,s)) / (\sum_{s \in S_r} |D(s)|)$.

We used a breadth-first search algorithm that is guaranteed to find the shortest path between two nodes in a graph if it exists [75]. Due to the huge size of vertices and edges in the baseline graph, it is necessary to implement some optimisation of the general breadth-first search. Since not all pairs of vertices in the graph are connected by a valid path and since the number of possible paths is huge, we defined a maximum path length and stopped

Table 2
Results of failure scenarios.

a. Top 20 ASs hosting most domain names

ASN	#domains	#res.	μ_{AS}	μ_{dom}	μ_{path}
AS20857	1,102,720	0	0	0	2.086
AS60781	910,200	0	0	0	2.049
AS21155	554,539	39	1	1	1.986
AS12859	554,182	0	0	0	2.065
AS8455	491,008	39	5	11,475	2.046
AS197902	458,068	0	0	0	2.015
AS25151	291,815	0	0	0	2.003
AS24940	248,358	0	0	0	2.022
AS48635	235,129	39	1	34,770	2.003
AS3265	234,512	0	0	0	2.004
AS35470	206,455	0	0	0	1.974
AS15879	199,021	0	0	0	1.998
AS25459	187,562	0	0	0	2.000
AS6724	172,155	39	1	0	2.004
AS49544	149,090	38[a]	4	1425	2.020
AS34233	148,421	0	0	0	1.997
AS61387	130,468	0	0	0	1.989
AS8315	113,760	0	0	0	2.002
AS50673	112,783	39	3	515	2.016
AS25525	99,106	39	1	15	2.001

b. Top 20 ASs providing most transit paths

ASN	#paths	#res.	μ_{AS}	μ_{dom}	μ_{path}
AS174	22,165	39	11	95	2.040
AS2914	17,849	39	4	3	2.037
AS1299	13,140	39	7	114	2.023
AS3356	12,932	15[a]	8	22	2.010
AS6453	6693	0	0	0	2.001
AS3320	6345	38[a]	2	1	2.007
AS20562	5011	1	1	0	2.049
AS6939	4861	39	5	10	2.011
AS43531	4374	39	1	0	2.007
AS10310	4026	39	102	137,560	1.932
AS4436	3990	2	1	0	2.000
AS49685	3946	39	101	178,110	1.923
AS8455	3716	39	5	11,475	2.046
AS5511	3649	14	110	83,677	2.107
AS16509	3592	14	108	83,359	2.059
AS9002	3473	39	1	0	2.001
AS8220	3192	39	10	29	2.001
AS1136	2997	39	41	51,101	2.106
AS3257	2982	7	1	0	2.002
AS701	2856	39	6	11	2.000

c. Top 20 ASs connections most traversed

connection	#paths	#res.	μ_{AS}	μ_{dom}	μ_{path}
AS35470-AS49685	3842	39	99	173,999	1.923
AS10310-AS36647	3842	39	99	137,559	1.932
AS1299-AS23033	3594	0	0	0	2.015
AS14618-AS16509	3594	14	108	71,124	2.048
AS3215-AS5511	3479	0	0	0	2.021
AS17204-AS2914	3128	0	0	0	2.016
AS2914-AS5432	3041	0	0	0	2.012
AS1136-AS8737	2826	0	0	0	2.028
AS31615-AS3320	2681	0	0	0	2.002
AS24793-AS41887	2657	14	107	59,305	2.004
AS1299-AS2637	2634	0	0	0	2.003
AS55967-AS6453	2396	0	0	0	2.001
AS30781-AS34173	2250	0	0	0	2.002
AS43531-AS9143	2225	1	3	1	2.004
AS2914-AS393406	2087	1	1	0	2.003
AS174-AS2914	2084	0	0	0	2.000
AS197902-AS8455	2047	0	0	0	2.003
AS3320-AS8972	1892	0	0	0	2.001
AS1136-AS286	1869	0	0	0	2.004
AS1299-AS13127	1800	0	0	0	2.005

[a] Actually one more resolver AS is affected, but this is left out since it coincides with the failing AS (see Section 4.9).

the search when paths contain more than 20 vertices. Ignoring paths of a certain length is justifiable as also the IP packets containing the DNS queries and responses on the network layer contain a Time To Live and do not traverse an infinite amount of machines before being discarded. Due to the high connectivity of the graph, most of the ASs are connected by multiple paths of length smaller than 6. A second optimisation is to mark a vertex once it is reached, which allows that a vertex has not to be considered again when it is encountered again in the search for a path. However, next to marking it is also necessary to store the relation by which the vertex was reached as some shorter paths may be invalid whereas a longer path using another relation to reach the vertex may be valid.

Table 1 shows the baseline results in column three, four and five. Since our topology is based on the CAIDA data that may be incomplete, and since we limited the maximum path length to 20, it is possible that not every AS is reachable from all other ASs in our topology. This indeed can be observed in column three of Table 1. In fact, one AS (AS26850) is not reachable from any resolver AS. This AS is very poorly connected and it just has one peering connection with one other AS. However, this only has a marginal impact on the analysis, as this AS just contains a single name server serving a single domain name in the .nl-zone.

Column five in Table 1 shows that the mean shortest path length varies between 1.182 and 2.886. In general it is not the case that ASs with a larger mean shortest path length are less well connected to the Internet. The mean shortest path length is weighted by the number of domain names, and hence paths to ASs hosting a large number of domain names contribute more to the mean than paths to ASs hosting only a small number of domain names. We chose to use the weighted mean shortest path length since this incorporates the reachability of DNS data, while using just the mean shortest path length would be an indicator for the general connectivity of ASs. This is also justified by Fig. 3, which shows that many ASs just host a single domain name and their impact on the reachability analysis should be less than the impact of ASs that host many domain names. The average of the mean shortest path lengths for all the 39 considered resolver locations (μ_{path}) is 2.000.

5.7. Identify most important ASs and interconnections

We identified the top 20 ASs in which most domain names are hosted (see Table 2.a, in which the first column lists the ASN and the second column lists the number of domain names hosted on name servers in the AS). We also identified the top 20 ASs that provide the most transit connections on paths in the baseline (see Table 2.b, in which the first column lists the ASN and the second column lists the number of paths in which the AS occurs as transit provider). One AS (AS8455) occurs in both lists.

We identified as well the interconnections between ASs that are most traversed in the baseline (see Table 2.c, in which the first column lists the interconnection and the second column lists the number of paths that include the interconnection).

5.8. Analyse DNS resilience with failures

We analysed the following failure scenarios:

- failure of an AS in the top 20 of ASs in which the most domain names are hosted;
- failure of an AS in the top 20 of ASs that provide the most transit connections on paths in the baseline;
- failure of an interconnection in the top 20 of the most traversed connections in the baseline;
- failure of the AMS-IX, implying that all peering interconnections at the AMS-IX fail.

We analysed the DNS resilience in each failure scenario by recomputing the reachability analysis on the failure graph in which the failing AS or interconnection is removed. The results are shown in the columns three to six of Table 2.

The third column (#res.) in the table shows the number of resolver locations affected. This number indicates the number of re-

solver locations out of the 39 most important resolver locations shown in Table 1, that are affected. Affected means that the number of unreachable ASs increases compared to the baseline (not counting the AS that is removed in the failure scenario).

The fourth column (μ_{AS}) and the fifth column (μ_{dom}) show the average number of ASs and the average number of domain names that become unreachable (where the mean is calculated over the affected resolver locations mentioned in the third column (#res)). In Table 2.a and 2.b the numbers in the fourth column (μ_{AS}) do not include the AS that is removed as it is clear that this AS will become unavailable for all other ASs. Likewise, the numbers in the fifth column (μ_{dom}) do not include the domain names which are hosted exclusively on name servers in the removed AS. Hence, these numbers are exclusive the ASs and domains which are expected to become unreachable (or already are unreachable in the baseline).

The sixth column (μ_{path}) shows the mean shortest path length in each failure scenario. This mean is calculated over all resolver locations, and hence it can be compared to the mean shortest path length (2.000) in the baseline.

In the following subsections we outline the main results. Detailed results for each failure scenario are available in [49].

5.8.1. Failure of ASs

The results for failure of ASs that host most domain names in Table 2.a indicate that in 13 of the 20 failure scenarios no resolvers are affected (except of course that the failing AS and the domain names hosted exclusively on this AS become unreachable). In the other 7 failure scenarios all resolver locations are affected, however still only small numbers of ASs become unreachable. Only the failures of AS8455 and AS48635 cause that a substantial amount of domain names becomes unreachable (which however still is a neglectable percentage of all domain names). Furthermore, the mean length of the shortest path to reach the AS hosting a domain name slightly increases for a subset of the resolvers. In some cases this mean length decreases a little, which is rather a side effect of calculating the mean over fewer domain names as some domain names are not reachable anymore. We conclude that failure of an AS that hosts many domain names in general has a minor impact.

The results for failure of ASs that provide most transit traffic in Table 2.b however show a different picture. Resolvers are affected in 19 of the 20 failure scenarios and larger numbers of ASs and domain names become unreachable. The reachability of domain names depends highly on the location of the resolver trying to access a certain domain name. For example in the scenario when AS10310 fails, AS36647 is completely cut off from the rest of the network and hence all 5.3 million domain names become unreachable for the resolvers in this AS. The same holds when AS49685 fails which causes that AS35470 is completely cut off. Also the cases when AS5511, AS16509 and AS1136 fail are interesting as in these cases the resolvers located in AS3215, AS14618 and AS8737, respectively, loose their connection to roughly half of the ASs where name servers are located, resulting in roughly 1.1 million unreachable domain names.

AS8455 is in both the top 20 of ASs serving most domain names and in the top 20 of ASs providing most transit traffic, and hence is an interesting case. Fig. 6 shows the ASs and domain names that become unreachable when AS8455 fails. The figure shows respectively: the number of ASs and domain names that were already unreachable from the 39 selected ASs in the baseline; AS8455 itself and the domain names served solely by AS8455 that become unreachable; and, the number of ASs and domain names that become unreachable in addition due to the failure of AS8455. Although the amount of additional unreachable ASs is limited, the amount of additional unreachable domain names is considerable. This indicates that some domain names can be resolved only by name servers that are all located in the same unreachable AS (or by just a single name server in an unreachable AS).

5.8.2. Failure of connections

The results for failure of connections between neighbouring ASs in Table 2.c show that removal of a single connection in general does not break anything as in 14 out of the 20 cases no ASs and thus no domain names other than those which are already unreachable in the baseline become unreachable.

In some situations however, the connection is very important for the reachability of certain ASs. AS35470-AS49685 and AS10310-AS36647 are the only connections for AS35470 and AS36647 to the rest of the network.

The failure of connection AS174-AS2914 does not have any impact at all, and even the mean shortest path length is unaffected. This would be the desired situation for all scenarios, as this suggests that the network offers alternative shortest paths of the same length.

5.8.3. Failure of the AMS-IX

We also analysed the failure of an IXP. We chose the AMS-IX since this is the largest IXP of the Netherlands and the majority of the Dutch ASs are members of the AMS-IX. In the failure scenario of AMS-IX we assume that all peering connections between the 769 member ASs are removed. Surprisingly, this leads to only a few changes in the amounts of unreachable ASs or domain names. In fact, the only resolvers affected are located in AS3320 (for which 116 ASs and 6096 domain names become unreachable) and in AS3356 (for which 57 ASs and 1406 domain names become unreachable). However, the mean length of the shortest path between the AS of the resolver and the AS of the name servers (μ_{path}) increases to 2.291, as shown in Fig. 7. The figure shows the mean length of the shortest path to reachable domains from the selected ASs containing resolvers, both for the baseline and in case AMS-IX fails. The increase in shortest path length is as expected, since now the traditional customer-provider links have to be utilised. In terms of pure reachability, the impact of the AMS-IX failing is thus quite small and only the mean shortest path length increases significantly. In practice however, there would still be availability issues as the available bandwidth of the remaining customer-provider links is not taken into account and these might become overloaded.

6. Discussion

To what extent our observations are realistic, is hard to judge. This is mainly due to the incompleteness of the underlying data regarding the AS topology. However, in our case study we did not observe any serious issues for the .nl-domain, and this observation would have been even stronger in case of more complete info on the AS topology. In case issues would have been observed, it would have been questionable how realistic these findings would have been. If this will happen when analysing other domains, more effort will be required to validate the findings. For instance, the data from looking glasses can be used as an additional source, but in case of deviations the question will remain which data source is more reliable.

A further limitation of our analysis is that we only consider logical connections between ASs and ignore physical links. ASs may be connected via multiple physical links which show up as a single logical connection. In case one such physical link fails, only the bandwidth of the connection would be affected, but the logical connection would still be in place. On the other hand, a single physical link may also bundle multiple logical connections, and in that case failure of a single link would imply that multiple logical

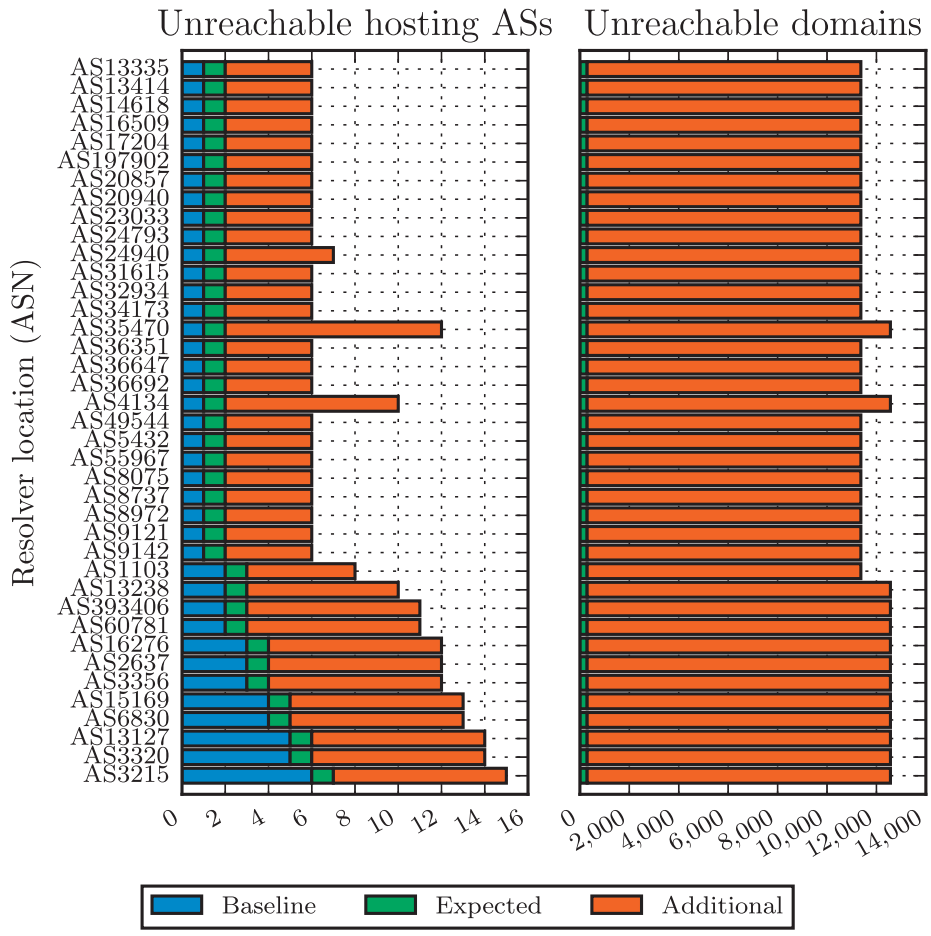

Fig. 6. Reachability of ASs and domain names when AS8455 fails.

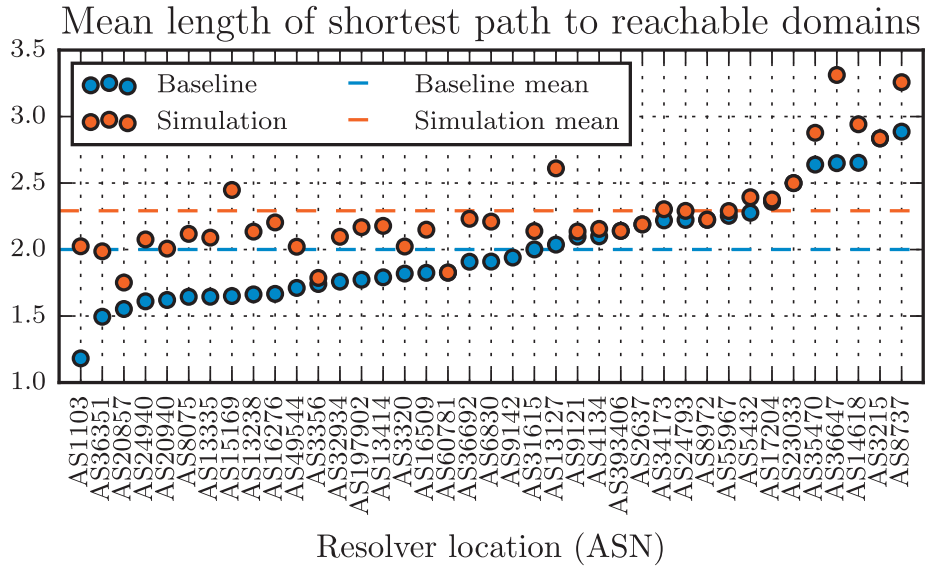

Fig. 7. Mean shortest path length from resolvers to reachable domain names when AMS-IX fails.

connections are affected. Also, a single logical connection may be physically distributed over two different networks in case of anycasting. We did not consider physical connections and bandwidth, mostly since there are no accurate, publicly available data on these, but it would be an interesting and valuable improvement of our method.

7. Conclusion

We presented a generic method to analyse resilience of the DNS infrastructure by investigating the impact when certain parts of the Internet's infrastructure fail. Although the method is hampered somewhat by the limited information about the Internet's infras-

tructure, we demonstrated in our case study that it can already be applied to a ccTLD and its second-level domains. An interesting follow-up study would be to compare the results obtained for the .nl-domain with other domains.

Our case study of the .nl-domain shows that the availability of most of the domain names in the .nl-zone is not at risk. This is largely in line with expectations. Our results show that most ASs have multiple connections to the Internet, and hence failure of a single connection has only a minor impact. However, we identified some ASs that are poorly connected, and also a substantial amount of domain names that are hosted solely on name servers within the same AS or in some extreme cases even on a single name server. Hence, these ASs and domain names are vulnerable and could become unreachable due to a single failure. Most of the issues occur only in extraordinary circumstances, when a complete AS fails, and can be circumvented by hosting the data on name servers located in different ASs.

References

[1] I. van Beijnum, BGP - Building Reliable Networks with the Border Gateway Protocol, 1st ed., O'Reilly Media, Inc., 2002.
[2] P. Mockapetris, Domain Names - Concepts and Facilities, Internet Engineering Task Force, 1987. RFC 1034.
[3] N. Chatzis, G. Smaragdakis, A. Feldmann, W. Willinger, There is more to IXPs than meets the eye, ACM SIGCOMM Comput. Commun. Rev. 43 (5) (2013) 19–28.
[4] R. Motamedi, R. Rejaie, W. Willinger, A survey of techniques for internet topology discovery, Commun.Surv.Tutor. IEEE 17 (2) (2015) 1044–1065.
[5] G. Huston, CIDR REPORT for 09 August 16, http://www.cidr-report.org/as2.0/.
[6] M. Caesar, J. Rexford, BGP routing policies in ISP networks, Netw. IEEE 19 (6) (2005) 5–11.
[7] Internet Assigned Number Authority, About the internet assigned number authority, https://www.iana.org/about.
[8] BBC, Cyber attacks briefly knock out top sites, http://www.bbc.com/news/technology-37728015.
[9] B. Al-Musawi, P. Branch, G. Armitage, BGP anomaly detection techniques: a survey, IEEE Commun. Surv. Tutor. 19 (1) (2017) 377–396.
[10] CNET, How Pakistan knocked YouTube offline (and how to make sure it never happens again), http://www.cnet.com/news/how-pakistan-knocked-youtube-offline-andhow-to-make-sure-it-never-happens-again/.
[11] Dyn, Pakistan hijacks YouTube, http://research.dyn.com/2008/02/pakistan-hijacks-youtube-1/.
[12] P. Mockapetris, Domain Names - Implementation and Specification, Internet Engineering Task Force, 1987. RFC 1035.
[13] International Organization for Standardization, Country codes - ISO 3166, http://www.iso.org/iso/country_codes/.
[14] J. Postel, Domain Name System Structure and Delegation, Internet Engineering Task Force, 1994. RFC 1591.
[15] Y. Rekhter, T. Li, S. Hares, A Border Gateway Protocol 4 (BGP-4), Internet Engineering Task Force, 2006. RFC 4271.
[16] V. Fuller, T. Li, Classless Inter-Domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan, Internet Engineering Task Force, 2006. RFC 4632.
[17] J. Rexford, J. Wang, Z. Xiao, Y. Zhang, BGP routing stability of popular destinations, in: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment, ACM, 2002, pp. 197–202.
[18] Internet Assigned Number Authority, Number resources, https://www.iana.org/numbers.
[19] C. Kruegel, D. Mutz, W. Robertson, F. Valeur, Topology-based detection of anomalous BGP messages, in: Recent Advances in Intrusion Detection, Springer, 2003, pp. 17–35.
[20] K. Zhang, A. Yen, X. Zhao, D. Massey, S.F. Wu, L. Zhang, On detection of anomalous routing dynamics in BGP, in: Networking 2004, Springer, 2004, pp. 259–270.
[21] C. Zheng, L. Ji, D. Pei, J. Wang, P. Francis, A light-weight distributed scheme for detecting IP prefix hijacks in real-time, in: ACM SIGCOMM Computer Communication Review, Vol. 37, ACM, 2007, pp. 277–288.
[22] L. Gao, On inferring autonomous system relationships in the internet, IEEE/ACM Trans. Network. 9 (6) (2001) 733–745.
[23] Center for Applied Internet Data Analysis, AS relationships, http://www.caida.org/data/as-relationships/.
[24] G. Shrimali, S. Kumar, Paid peering among internet service providers, in: Proceeding from the 2006 Workshop on Game Theory for Communications and Networks, ACM, 2006, p. 11.
[25] P. Faratin, D.D. Clark, S. Bauer, W. Lehr, P.W. Gilmore, A. Berger, The growing complexity of internet interconnection, Commun.Strat. (72) (2008) 51.
[26] A. Dhamdhere, C. Dovrolis, P. Francois, A value-based framework for internet peering agreements, in: Teletraffic Congress (ITC), 2010 22nd International, IEEE, 2010, pp. 1–8.
[27] S. Lippert, G. Spagnolo, Internet peering as a network of relations, Telecomm. Policy 32 (1) (2008) 33–49.
[28] M. Roughan, W. Willinger, O. Maennel, D. Perouli, R. Bush, 10 lessons from 10 years of measuring and modeling the internet's autonomous systems, Sel.AreasCommun. IEEE J. 29 (9) (2011) 1810–1821.
[29] European Internet Exchange Association, What is an IXP?, https://euro-ix.net/ixps/what-is-ixp/.
[30] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, W. Willinger, Anatomy of a large European IXP, in: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, ACM, 2012, pp. 163–174.
[31] I. van Beijnum, Transit vs peering: what makes sense when?, http://packetpushers.net/transit-vs-peering-makes-sense/.
[32] Amsterdam Internet Exchange, Ams-IX route servers, https://ams-ix.net/technical/specifications-descriptions/ams-ix-route-servers.
[33] E. Basturk, R. Engel, R. Haas, V. Peris, D. Saha, Using Network Layer Anycast forLoad Distribution in the Internet, Technical Report, IBM T.J. Watson Research Center, 1997.
[34] H. Miura, M. Yamamoto, K. Nishimura, H. Ikeda, Server load balancing with network support: active anycast, in: Active Networks, Springer, 2000, pp. 371–384.
[35] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, Two days in the life of the DNS anycast root servers, in: Passive and Active Network Measurement, Springer, 2007, pp. 125–134.
[36] S. Sarat, V. Pappas, A. Terzis, On the use of anycast in DNS, in: Computer Communications and Networks, 2006. ICCCN 2006. Proceedings. 15th International Conference on, IEEE, 2006, pp. 71–78.
[37] L. Colitti, E. Romijn, H. Uijterwaal, A. Robachevsky, Evaluating the effects of anycast on DNS root name servers, RIPE Document RIPE-393, 2006.
[38] D. Magoni, J.J. Pansiot, Analysis of the autonomous system network topology, ACM SIGCOMM Comput. Commun. Rev. 31 (3) (2001) 26–37.
[39] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, A. Vahdat, The internet AS-level topology: three data sources and one definitive metric, ACM SIGCOMM Comput. Commun. Rev. 36 (1) (2006) 17–26.
[40] B. Zhang, R. Liu, D. Massey, L. Zhang, Collecting the internet AS-level topology, ACM SIGCOMM Comput. Commun. Rev. 35 (1) (2005) 53–61.
[41] Y. Zhang, Internet AS-level topology archive, http://irl.cs.ucla.edu/topology/.
[42] L. Subramanian, S. Agarwal, J. Rexford, R.H. Katz, Characterizing the internet hierarchy from multiple vantage points, in: INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 2, IEEE, 2002, pp. 618–627.
[43] S. Trajanovski, F.A. Kuipers, A. Ilić, J. Crowcroft, P.V. Mieghem, Finding critical regions and region-disjoint paths in a network, IEEE/ACM Trans. Network. 23 (3) (2015) 908–921.
[44] H. Haddadi, M. Rio, G. Iannaccone, A. Moore, R. Mortier, Network topologies: inference, modeling, and generation, Commun. Surv. Tutor. IEEE 10 (2) (2008) 48–69.
[45] N. Brownlee, K. Claffy, E. Nemeth, DNS measurements at a root server, in: Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE, Vol. 3, IEEE, 2001, pp. 1672–1676.
[46] L. Wei-Min, C. Lu-Ying, L. Zhen-Ming, Alleviating the impact of DNS DDos attacks, in: Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on, Vol. 1, IEEE, 2010, pp. 240–243.
[47] W.M. Li, X.G. Cao, F. Liu, Z.M. Lei, Improving DNS cache to alleviate the impact of DNS DDos attack, J. Netw. 6 (2) (2011) 279–286.
[48] G.C. Moura, R. de Oliveira Schmidt, J. Heidemann, W.B. de Vries, M. Müller, L. Wei, C. Hesselman, Anycast vs. DDos: evaluating the november 2015 root DNS event, in: Proceedings ACM Internet Measurement Conference, ACM, 2016, pp. 255–270.
[49] L. Bade, Resilience of the Domain Name System: A Case Study for .nl, Radboud University, 2016 MSc Thesis.
[50] SIDN, Spark: scan the DNS efficiently, https://github.com/SIDN/spark.
[51] L. Deri, S. Mainardi, M. Martinelli, E. Gregori, Exploiting DNS traffic to rank internet domains, in: Proceedings IEEE International Conference on Communications Workshops (ICC), IEEE, 2013, pp. 1325–1329.
[52] Center for Applied Internet Data Analysis, The CAIDA AS organizations dataset, http://www.caida.org/data/as-organizations.
[53] L. Daigle, WHOIS Protocol Specification, Internet Engineering Task Force, 2004. RFC 3912.
[54] SIDN Labs, ENTRADA - an open source platform for network data analytics, http://entrada.sidnlabs.nl.
[55] University of Twente and SURFnet and SIDN, OpenINTEL open access, http://openintel.nl/.
[56] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, A. Pras, A high-performance, scalable infrastructure for large-scale active DNS measurements, IEEE J. Sel. Areas Commun. 34 (6) (2016) 1877–1888.
[57] S. Bortzmeyer, DNS Query Name Minimisation to Improve Privacy, Internet Engineering Task Force, 2016. RFC 7816.
[58] University of Oregon, Route views project, http://www.routeviews.org/.
[59] RIPE Network Coordination Centre, Routing information service (RIS), https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.
[60] Center for Applied Internet Data Analysis, Archipelago (Ark) measurement infrastructure, http://www.caida.org/projects/ark/.

[61] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, G. Riley, AS relationships: inference and validation, ACM SIGCOMM Comput. Commun. Rev. 37 (1) (2007) 29–40.
[62] X. Dimitropoulos, D. Krioukov, B. Huffaker, G. Riley, Inferring AS relationships: dead end or lively beginning? in: Experimental and Efficient Algorithms, Springer, 2005, pp. 113–125.
[63] V. Giotsas, S. Zhou, M. Luckie, Inferring multilateral peering, in: Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, ACM, 2013, pp. 247–258.
[64] RIPE Network Coordination Centre, ASN neighbours, https://stat.ripe.net/widget/asn-neighbours.
[65] D. Stamatis, Failure Mode and Effect Analysis: FMEA from Theory to Execution, ASQC Press, 1995.
[66] R.R. Dye, Identification and classification of modes of failure in aeronautical equipment, IEEE Trans. Aerosp. 2 (2) (1964) 535–542.
[67] I. Elyasi-Komari, A. Gorbenko, V. Kharchenko, A. Mamalis, Analysis of computer network reliability and criticality: technique and features, Int. J. Commun. Netw. Syst. Sci. 4 (2011) 720–726.
[68] V. Pappas, P. Fältström, D. Massey, L. Zhang, Distributed DNS troubleshooting, in: Proceedings of the ACM SIGCOMM Workshop on Network Troubleshooting: Research, Theory and Operations Practice Meet Malfunctioning Reality, ACM, 2004, pp. 265–270.
[69] MaxMind Inc, Geolite legacy downloadable databases, http://dev.maxmind.com/geoip/legacy/geolite/.
[70] RIPE Network Coordination Centre, RIPE database, https://apps.db.ripe.net/search/query.html.
[71] Team Cymru Inc, IP to ASN mapping, http://www.team-cymru.org/IP-ASN-mapping.html.
[72] SIDN, Technical requirements for the registration of.nl domain names, https://www.sidn.nl/downloads/terms-and-conditions/Technical+requirements+for+the+registration+of+nl+domain+names.pdf (2010).
[73] Google, Verifying Googlebot, https://support.google.com/webmasters/answer/80553.
[74] Google, Google public DNS - performance benefits, https://developers.google.com/speed/public-dns/docs/performance.
[75] S.S. Skiena, The Algorithm Design Manual: Text, Vol. 1, Springer Science & Business Media, 1998.

Lars Kröhnke (birth name Lars Bade) received a M.Sc. on computer science from the Radboud University Nijmegen in 2016, after he finished his Master's Thesis research on the topic of this paper as an intern at SIDN Labs. During the study period he specialised in the field of computer security by following a special master program provided by the Kerckhoffs Institute. Since he left the university he is employed as a Technical Consultant at GX Software in Nijmegen, the Netherlands.

Jelte Jansen received a M.Sc. on computer science from the Radboud University Nijmegen in 2004. He is currently a research engineer at SIDN Labs, the research and development team of SIDN, where his research and development topics include the domain name system, internet protocols, and privacy/identity management. Jelte is an editor for Privacy & Informatie, and a member of the programme committee at RIPE meetings. He is also member of the IETF, where he works on standardization of technologies such as DNS and DNSSEC, and has worked on several DNS implementations as a software developer.

Harald Vranken received a M.Sc. on information technology in 1992, a PDEng. on information and communication technology in 1994, a Ph.D. on electrical engineering in 1998, and a M.Sc. on science education and communication in 2006, all from the Eindhoven University of Technology. He is currently an Associate Professor at the Open University and Radboud University in The Netherlands. His current research interests include resilience, software security, and network security. He was a senior scientist at Philips Research Labs from 1998 to 2005 where a worked on design-for-testability of digital ICs.