Graph Theoretical Models of DNS Traffic

Luca Deri*, Simone Mainardi*†, Maurizio Martinelli*
and Enrico Gregori*
*Institute of Informatics and Telematics (IIT), Italian National Research Council (CNR), Pisa, Italy.
Email: firstname.lastname@iit.cnr.it
†Department of Information Engineering (IET), University of Pisa, Pisa, Italy.
Email: simone.mainardi@iet.unipi.it

***Abstract*—The DNS is one of the core protocols on which the Internet is built upon. Hidden behind higher-level protocols such as email and web, it carries valuable information that can be exploited for understanding trends and preferences of the Internet community.**

In this paper we propose novel methodologies for modelling DNS traffic that allow Internet domains, DNS resolvers and their interactions to be represented effectively by means of graphs. DNS traffic collected at ".it" ccTLD DNS domain servers has been used to validate this work on a large scale. We found highly-skewed, fat-tailed domain and resolver degree frequencies, obeying power laws at least in their tails. These findings shed light on the the scale-free nature of the DNS ecosystem, where a few domains and a few resolvers are responsible for most of the DNS activity.

I. Introduction and Related Work

The domain name system (DNS) is a an essential component of the Internet used to associate symbolic host names with numeric IP addresses. Internet service providers often perceive the DNS as a core system they must keep up and running as their customers rely on it, but being it a service that does not bring revenues, they do not usually invest much on it. The consequence is that ISP's DNS servers are sometimes slow in responses [1], and this has opened the market to public DNS servers such as OpenDNS and Google Public DNS. Beside premium services, such public DNSes offer the service at no cost while making revenues through advertisements, web traffic redirection and mining of DNS data. Although the DNS is perceived as a critical infrastructure [2], all publicly available DNS traffic monitoring tools [3] [4] focus only on aggregate values such as the type and number of queries received by a DNS server [5]. Research and academia have focused on DNS for the purpose of identifying malicious activities [6] [7] [8], managing large DNS infrastructures [9], understanding how DNS server selection and caching works in reality [10] [11], and modeling its infrastructure in order to predict how DNS traffic will change under specific conditions [12]. The lack of specific models for DNS traffic has been the motivation for this work [13] [14]. As explained later on this paper, such models can be of fundamental importance for understanding patterns, trends and interests in the Internet without having to monitor large amount of (often encrypted) traffic on multi-Gbit backbones.

The rest of this paper is organized as follow. Section II describes in detail domain name resolution and record caching in the DNS system. Section III covers in detail the graph theoretical methodologies we propose to model DNS traffic. Section IV describes the results we have obtained while applying our methodology to the monitoring of ".it" ccTLD authoritative DNS servers. Finally, in Sect. V we identify some future work items. We conclude the paper with a brief summary in Sect. VI.

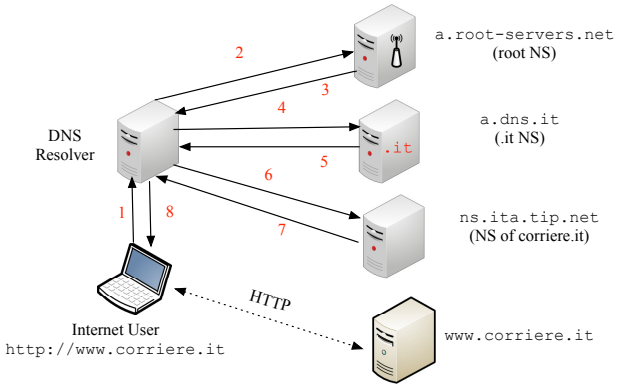

Fig. 1. Steps in DNS Resolution of `www.corriere.it`

II. Understanding The DNS System

A. Iterative Domain Name Resolution

The domain name system is based on a hierarchical distributed architecture used to map domain names to resource records containing various types of data including numeric IP addresses (A record for IPv4 and AAAA for IPv6), names (NS record) and mail exchange servers (MX records) [8]. The DNS resolver is the client side of the DNS system, responsible for performing address resolution by starting from the top of the hierarchy (i.e., the *root zone*). The root zone is served by 13 root name servers, most of which with anycast addressing [15]. The address resolution process is iterative and involves contacting several name servers, each one responsible (i.e., *authoritative*) for a part of the domain name. In Fig. 1 we show an example. The process is triggered by an Internet user which, with the aim of establishing an HTTP session, queries a DNS resolver to obtain the IP address of `www.corriere.it`. The DNS resolver sequentially contact three name servers: the first (`a.root-servers.net`)

replies indicating a.dns.it as authoritative name server for .it domain names; the second (a.dns.it) replies indicating that ns.ita.tip.net is authoritative for domain names ending with corriere.it; the third (ns.ita.tip.net), authoritative for www.corriere.it replies with the address of the host. Once the DNS resolver has received the IP address from the last authoritative server, it send it to the user which eventually establishes an HTTP session with www.corriere.it.

B. DNS Records Caching

Each record has a Time To Live (TTL) [16], that can range from 0 (i.e., no cache) to days or weeks. It determines for how long the given response record can be kept in cache. The consequence of the DNS caching architecture is that DNS record updates do not propagate immediately in the network until cached records expire. Record caching is a pretty complex mechanism [16] as all DNS records used in the resolution process do not necessarily have uniform TTL values. Supposing that a DNS resolver starts with an empty cache the resolution of www.corriere.it, its cache at the end of the iterative process will be populated with several records – each one with its own TTL. For example, the A record of www.corriere.it has TTL equal to 600 seconds, shorter than the NS record of corriere.it (10,800 seconds) and shorter than the NS record of .it (172,800 seconds). So if www.corriere.it is requested after 700 seconds, the DNS resolver will no longer have its IP in cache as the A record expired in the meantime. However, it will still have in cache the NS records for corriere.it and .it. The resolver will contact again ".it" DNS servers only after the NS record for corriere.it has expired[1].

To make caching even more complex to understand, differences in DNS implementations must also be taken into account. In the above example, the NS record for corriere.it has a TTL of 10,800 seconds as it has been set by all the ".it" DNS servers, but as its TTL reported by the authoritative DNS of corriere.it is 600 seconds (i.e., dig -t NS corriere.it ns.ita.tip.net) some DNS implementations might override 10,800 with 600, making harder to predict the resolver cache contents.

C. Monitoring DNS Traffic

As previously explained, we are monitoring the DNS traffic at the ".it" DNS servers. This means that we only observe queries for .it (i.e., we do not observe queries for .org or .net) and only for those domains for which ".it" DNS servers are authoritative. Out of all the DNS traffic we observe, in the methodology described later in this paper, we take into account only the AAAA, A, and MX records. In addition, for A and AAAA records, we ignore queries for both hosts that are known to be DNS servers and for which ".it" DNS servers are not authoritative (e.g. A record of www.sub-domain.domain.it). The reasons why we discard these queries are manyfold:

- Records other than A, AAAA, MX are used by the DNS infrastructure to resolve addresses (e.g. the NS record) or as ancillary records (e.g. PX records).
- A and AAAA record queries (as well NS records) for hosts that are DNS servers should not be taken into account because:
  - They have been requested due to the DNS caching mechanism where A records of DNS servers expire at different times than the corresponding NS record.
  - DNS servers are usually authoritative for many domains, and thus whenever we observe a A/AAAA record query for a DNS server, it is not possible to associate it with the domain name for which it has been requested.

In conclusion, we do not need to account all DNS queries but only those that indicate user activity such as an MX record query that indicate that an email will be sent, or A and AAAA records of hosts other than DNS servers (e.g. www.nic.it) that instead are used by the DNS system to resolve addresses. Please note that in theory, for a given resolver, once a record has been put in cache, no further query for the same record will be issued before the record expires as specified by the TTL. In practice this does not always hold, as most resolvers select authoritative name servers based on their response time. This explains why we often observe some extra queries used by resolvers to estimate the response time of all authoritative DNS servers for a given Internet domain. Thus beside these probing queries, we can identify resolvers that do not obey to the DNS specification when they perform queries that are well above the limit set by the TTL for the specified record. It is worth to remark, that resolvers identified using this method, cannot always be considered as malicious hosts. This is because sometimes network administrators periodically flush resolvers caches in order to reduce memory usage and thus extra queries are observed. For this reason we mark resolvers as malicious only whenever they significantly exceed the number of queries specified by the TTL.

III. DNS Traffic Modelling Methodology

In this section we first describe the rationale behind the assumptions made to compare domains and resolvers with non-uniform configurations. Then we explain the graph theoretical models developed, after briefly introducing some elements of graph theory.

A. Normalizing Non-Uniform TTL Values

Goal of our monitoring methodology is to passively collect DNS request/responses in order to mimic the DNS cache of resolvers that contacts ".it" DNS servers [17]. In order to model DNS traffic, we need to take into account the TTL and not just count DNS queries. In fact, if domain A has a TTL greater than domain B, a resolver that has to resolve both A and B addresses continuously throughout the day, will issue fewer queries for A than B, as A records have a longer

[1]It is worth to remark a name server can have different TTL values for NS records of domains it is authoritative for. Thus even within a single domain, TTLs might not be necessarily uniform.

cache lifetime than B records. Since we need to deal with all the ".it" domains, our methodology should enable domains with heterogeneous TTL values to be compared. This means that TTLs have to be normalized to the maximum TTL value among all the observed TTL values for NS records. As we measured that less than 2% of – about 2.5 million at the date of writing – ".it" domains use a TTL greater than 86,400 sec (1 day), we decided to use 1 day as baseline for our graph theoretical DNS models.

B. Elements of Graph Theory

A *graph* $G = (V, E)$ is a pair of sets (V, E), with V the set of $n = |V|$ nodes and $E \subseteq V \times V$ the set of edges. If edges $(i, j) \in E$ are unordered pairs, then G is said to be *undirected*. Two nodes $i, j \in V$, $i \neq j$, are said to be adjacent if $(i, j) \in E$. Adjacency relationships can be represented for each pair of nodes i and j $(i, j = 1, \cdots, n)$ with an n-square adjacency matrix $\mathbf{A}$ whose off-diagonal elements $a_{i,j}$ are equal to 1 if $(i, j) \in E$ or 0 otherwise — as self-edges are not allowed, in-diagonal elements $a_{i,i}$ are equal to 0. The *degree* $d_i = \sum_j a_{i,j}$ of a node i is the number of nodes adjacent with i. $\mathbf{A}$ can be generalized by associating real numbers to its elements $a_{i,j}$ in order to encode general tie strengths between nodes rather than binary adjacencies. In the latter case the graph is said to be *weighted*. Whenever V can be divided into two disjoint sets R and D such that each edge joins a node in R to a node in D, then G is said to be *bipartite*. Any bipartite graph can be represented with an adjacency matrix of the form

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix},$$

where $\mathbf{B} = [b_{r,d}]$ is a matrix with $|R|$ rows and $|D|$ columns, uniquely identifying the bipartite graph. Rows (columns) of $\mathbf{B}$ represent nodes in R (in D) and elements $b_{r,d}$ are equal to 1 whenever r and d are adjacent or 0 otherwise.

C. Bipartite Graph Model of the DNS

A very effective way of modelling resolvers, domains and their interactions is through an *undirected bipartite graph* $G = (V, E)$, such that $V = R \cup D$ and $R \cap D = \emptyset$. We take as R the set of resolvers and as D the set of domains. Bipartite graphs of the DNS can be built by placing edges between a resolver $r \in R$ and a domain $d \in D$ whenever a certain condition is met. In this paper we generate the following bipartite graphs:

- G_{ALL}: we place an edge between r and d iff r issued at least one DNS query for d in the observation period for A, AAA and MX records.
- G_{WEB}: we place an edge between r and d iff r issued at least one DNS query for d in the observation period for A and AAAA records and specify a name which is: the domain name with no host specified (e.g. `nic.it`); or the domain name preceded by either `www` or `web` (e.g. `www.nic.it`). In essence, we consider only those DNS queries that should be originated uniquely by web traffic[2].
- G_{MX}: we place an edge between r and d iff r issued at least one DNS query for d in the observation period for MX records. In essence we consider only `.it` Internet domains email traffic[3].

In all the previous cases, once the condition is met, the degree k_r of a resolver $r \in R$ equals the number of domains it issues queries for. Similarly, k_d of a domain $d \in D$ represents the number of resolvers that query a given domain name. Since we excluded isolated nodes, i.e. resolvers and domains with zero degree, we have that degrees are always not less than one.

D. Common-Neighbours Graph Model of the DNS

Internet domains and DNS resolvers can also be modelled in a way that the concept of adjacency becomes associated to the number of their common neighbours. In the case of two domains d_1 and d_2, *common neighbours* represent DNS resolvers which issue queries for both d_1 and d_2. This value is obtained by multiplying element-by-element columns of $\mathbf{B}$ with indices d_1 and d_2 and summing the result, i.e. $\sum_{r=1}^{|R|} b_{r,d_1} b_{r,d_2}$. Equivalently, the number of *common resolvers* can be directly computed for any pair of domains by taking the matrix product $\mathbf{B}^T\mathbf{B}$. Assuming the latter product is an adjacency matrix of a weighted graph $G_D = (D, E_D)$, then it turns out that G_D has D as its set of nodes. In addition, any of its edges $(d_1, d_2) \in E_D$ is associated with a positive weight corresponding to the number of resolvers shared by d_1 and d_2. Similarly, the number of *common domains* for each pair of resolvers can be obtained by computing $\mathbf{B}\mathbf{B}^T$, which in turn can be taken as the adjacency matrix of a weighted graph $G_R = (R, E_R)$ having R as its set of nodes. Each edge $(r_1, r_2) \in E_R$ is associated with a positive weight corresponding to the number of domains shared by r_1 and r_2.

IV. Results and Validation

The ".it" zone has seven administrative DNS servers, three of which with anycast addresses. The ".it" DNS monitoring system [18] we used for validating this work monitors four authoritative name servers. Two name servers have anycast address (`a.dns.it` located in Rome and Milan) and two have unicast address (`dns.nic.it` and `nameserver.nic.it`, both located in Pisa). Every ".it" DNS server node serves about 40 million requests/day, and we passively collect DNS traffic

[2]We are aware that using this approach, some web traffic might not be accounted. This happens whenever a resolver 1) issues queries for hosts that are not marked as web although they are in practice a web site (e.g. `video.mysite.it`) or 2) has issued a query for a record other than `www` (e.g. `mx.nic.it`) prior to issue a query for `www`. In this case, since the resolver cache was already filled-up, the query for `www` could not be observed. We estimate these are few cases as the probability that a web user makes non-web activity with a domain prior to access the web site is small.

[3]It is worth to remark, that in case a domain name has no MX record defined, email senders query the A record of the domain name. This means that queries for the exact domain name can either be due to emails or web traffic. In general as most domains have the MX record defined, we account queries for exact domain name into web traffic.

TABLE I
STRUCTURAL PROPERTIES OF BIPARTITE DNS GRAPHS

Property	G_{ALL}	G_{WEB}	G_{MX}
no. of edges	33,047,646	17,896,183	2,099,030
no. of domains	1,779,249	1,693,180	323,956
no. of resolvers	511,039	294,091	99,016
avg. d. degree	18.75	10.57	6.48
avg. r. degree	64.67	60.86	21.20

TABLE II
STRUCTURAL PROPERTIES OF COMMON-NEIGHBOURS DNS GRAPHS

Property	G_D	G_R
no. of edges	124,717	124,750
no. of nodes	500	500
avg. edge weight	1253.72	1972.97
avg. node intensity	625,442.89	984,516.46

using a home-grown open-source NetFlow probe [19] featuring a plugin for dissecting DNS query/responses. This solution allows us to be independent from the DNS implementation being used, and thus be general enough to use it on different contexts. The system is producing daily domain ranking since Jan 2013. In this section we present the monitoring results we observed on Jan 4th, 2013 while monitoring `dns.nic.it`. Every night we consolidate the DNS traffic logs produced by the probe, and produce:

- The G_{ALL}, G_{WEB}, G_{MX} graphs.
- The domain and resolver rankings based on score we described on this paper.
- The consolidated list of suspicious DNS activities carried on by resolvers that we use to spot potential issues.

A. Structural Properties of DNS Graphs

Structural properties of bipartite graphs G_{ALL}, G_{WEB}, G_{MX} are reported in Tab. I. For each graph we indicate: the number of edges $|E|$; the number of domains $|D|$ and resolvers $|R|$ with degree greater than zero; the average domain degree $\overline{d_D} = |D|^{-1} \sum_{d \in D} d_d$; and the average resolver degree $\overline{d_R} = |R|^{-1} \sum_{r \in R} d_r$.

We observe that G_{WEB} has half the edges of G_{ALL}. In addition, while the number of domains is slightly smaller, resolvers reduce significantly (from more than 500,000 to less than 300,000), highlighting that a large part of them is not interested in resolving names for web resources – although web traffic may be under-estimated as previously described in Sect. III-C. On average, resolvers query the same number of domains in G_{ALL} and G_{WEB} as highlighted by values of average resolver degree which are pretty close in both cases. Each domain is resolved for its web resources only one half of the times as the average domain degree halves from G_{ALL} to G_{WEB}. Much more significant reductions are observed for structural properties of graph G_{MX}, suggesting that only the 20% of resolvers issue queries uniquely for mail. Although this fact deserves further investigation, it may indicate that one resolver out of five has mail servers among its users.

We now deepen the analysis by further studying node degrees. In Fig. 2(a) we show Log-Log plots of frequency f_d versus degree d for domains and resolvers. *Frequency* f_d is the number of domains (resolvers) having degree d. These plots span approximately 5 orders of magnitude, indicating great variability and skewness in the degree of domains and resolvers. We observe that a single resolver can issue queries for more that 100k different domains, while the maximum number of resolvers querying a domain never exceed 50k. Nevertheless, these are statistically rare events. Indeed, we observe that domains with a low number of queries are the most frequent. Similarly, resolvers querying a low number of domains are the most common. Hence, the *"interest"* shown for Internet domains from DNS resolvers is far from being arbitrary. This "interest" can be quantified to some extent by observing that, for some ranges of d, plots are approximately linear in the Log-Log plots of Fig. 2. Linearity in the log-log scale mean that node degree follows a *power law*. Mathematically, a function $f(x)$ follows power law if it varies proportionally to a power γ of x, i.e., $f(x) \propto x^{\gamma}$. In our case, power law means that domain (resolver) i has a chance of having degree d_i which is proportional to the degree raised to a constant power γ. We estimated γ values using the method described in [20], which also provides values d_{min} such that power laws are obeyed only for $d \geq d_{min}$. Power laws fits with best γ values are plotted in Fig. 2 as dark lines, starting from d_{min} for each degree frequency. Tails of domain degree frequency in both G_{ALL} and G_{WEB} follow very well power laws with strikingly close exponents. In G_{MX} the power law is obeyed for a much lower d_{min}. Similarly, power laws are observed for resolver degree frequencies but with lower exponents, indicating slower decays. Since power laws with exponent less that -2 have infinite mean and variance [21], we stress on the extreme skewness of resolver degree. Nevertheless, also resolvers in G_{ALL} and G_{MX} degree obey power laws with very close exponents. We also compared our power law estimations with other distributions through *likelihood ratio tests* [20]. Such tests suggested that frequencies may also be well approximated by stretched exponentials and truncated power laws, but they definitely excluded gamma, log-normal and exponential distributions. Further investigations are left as future work. Nevertheless, we can conclude that a few domains (resolvers) are responsible for most of the DNS activity and power laws well describe this activity.

We have also analysed the common-neighbours DNS graphs. We generated weighted graphs G_D and G_R by selecting the top 500 highest-degree domains (resolvers) and all their adjacent resolvers (domains) from G_{ALL}. In both cases, the weight of each edge equals the number of neighbours in common. Structural properties of common-neighbours graphs G_D and G_R are reported in Tab. II. Both graphs have a number

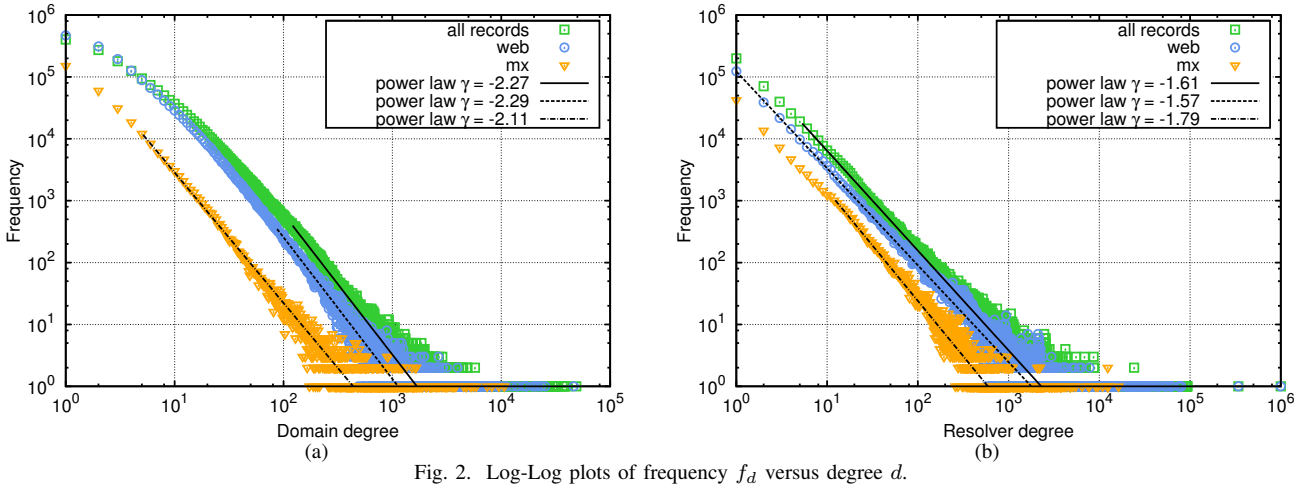

Fig. 2. Log-Log plots of frequency f_d versus degree d.

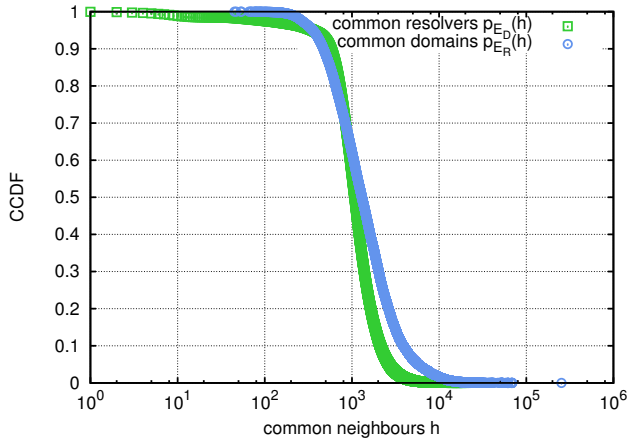

Fig. 3. G_D and G_R common neighbours CCDF $p_{E_D}(h)$ for domains and $p_{E_R}(h)$ for resolvers.

TABLE III
DOMAIN SCORE COMPARISON

Domain Name	dns.nic.it	a.nic.it	Difference
amazon.it	1	1	=
telecomitalia.it	2	4	+2
virgilio.it	3	3	=
fastwebnet.it	4	2	−2
corriere.it	5	5	=
tiscali.it	6	6	=
aruba.it	7	5	−2
google.it	8	11	+3
vodafone.it	9	12	+3
gazzetta.it	10	8	−2

of edges which approaches the maximum, hence any pair of domains (resolvers) have at least one neighbour in common. By observing the average edge weight, it turns out that pairs of domains on average are requested by more than 1200 common resolvers. Similarly, pairs of resolvers on average query approximately 2000 domains. In addition, average node intensities (i.e., the average of the sum of edge weights for each node) tell that a large number of resolvers (domains) is shared among top 500 highest-degree domains (resolvers).

In Fig. 3 we show the Complementary Cumulative Distribution Function (CCDF) $p_{E_D}(h) = P(w_{d_1,d_2} > h|(d_1,d_2) \in E_D)$ giving the probability that any pair of domains $(d_1,d_2) \in E_D$ has a number w_{d_1,d_2} of resolvers in common greater that h as function of h. Similarly, we show the CCDF $p_{E_R}(h) = P(w_{r_1,r_2} > h|(r_1,r_2) \in E_R)$ which gives the chance that any pair of resolvers $(r_1,r_2) \in E_R$ shares a number of domains w_{r_1,r_2} greater that h. Both distributions are almost identical up to 1000, whereas the resolvers start decaying much faster than domains.

B. Results Evaluation

We have also compared the degree of top 100 domains on two monitored name servers (`dns.nic.it` and `m.dns.it`), assigning a score from 1 to 100 based on the domain degree. We have found that over 93% of domains are present on both servers, and that the domain scores are pretty close but not alike as shown in Tab. III. Resolvers score have instead a different behaviour:

- Resolver score are very different across the two name servers servers as only 52% of top 100 resolvers contacted both servers.
- For each name server, the list of top resolvers across various days reports only minor differences.

As the Round Trip Time (RTT) is the metric used to choose between authoritative servers for the same zone [22] [23], not all name servers are alike for resolvers as they prefer those that reply with lower RTT[4]. This is also justified by a probing activity that we have identified in our traffic traces, where several resolvers issue the same query to all monitored ".it" authoritative name servers twice in about 10 seconds, in order to figure out which one reports the lowest RTT; this will be

[4]BGP (Border Gateway Protocol) peering allows RTT to be reduced for those resolvers belonging to ASs (Autonomous Systems) for which there is a peering relations, thus affecting the resolvers distribution across monitored name servers. This is because depending on the site where the name server is located, there are non-uniform peering relationships in place.

the selected name server to be used for future queries on that domain name.

V. Future Work Items

The described methodology is a good starting point for several follow up activities including:

- Exploit the neighbourhood relation shown in Fig. 3 to highlight relationships in web traffic. For instance the number of resolvers in common between two domains can be used to:
  - Evaluate the effectiveness of advertisements (e.g. restricted to web traffic, counting the number of resolvers in common between `X.it` and `companyname.it` allows to figure out on which web sites `companyname.it` places its advertisements).
  - Understand the chain of interests of people (e.g. Internet users who access news site `X` will likely also access news site `Y`).
- In order to have an efficient DNS infrastructure, the goal is to minimize the RTT for resolvers. Using our models, we can start optimizing traffic routing so top scoring resolvers can be reached by `.it` name servers with a low TTL thus resulting on a more efficient service.
- Refinement of the algorithm used to identify resolvers that do not obey to the TTL, so that we can distinguish between misbehaving or misconfigured resolvers worth to track as they may conduct malicious activities [24], and resolvers running malfunctioning DNS implementations that instead should not be taken into account.

VI. Conclusion

This paper describes novel methodologies for modelling DNS traffic by means of graphs. During the validation on the ".it" ccTLD authoritative name servers, we found that DNS resolvers degrees are highly-skewed and obey power laws. This fact also holds for Internet domains when considering all traffic or just a portion of it such as web or email. These findings give new insight into the scale-free nature of the DNS system, where a few resolvers and a few domains are responsible for most of the DNS activity.

Acknowledgement

The authors would like to thank out colleagues at CNR-IIT and in particular Stefano Ruberti and Daniele Vannozzi for their help and support throughout this project.

References

[1] G. R. Corporation, "Domain name speed benchmark," 2012. [Online]. Available: https://www.grc.com/dns/benchmark.htm

[2] E. Casalicchio and I. N. Favino, "The 1st workshop on dns health and security," 2012. [Online]. Available: http://www.gcsec.org/sites/default/files/files/dnseasy2011.pdf

[3] D. Plonka and P. Barford, "Context-aware clustering of dns query traffic," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008.

[4] T. M. Factory, "dnstop and dsc tools," 2006. [Online]. Available: http://dns.measurement-factory.com/tools/

[5] E. Osterweil, D. McPherson, S. DiBenedetto, C. Papadopoulos, and D. Massey, "Behavior of dns' top talkers, a .com/.net view," in *Proceedings of the 13th international conference on Passive and Active Measurement*. Berlin, Heidelberg: Springer-Verlag, 2012.

[6] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for dns," *9th Usenix Security Symposium*, 2010.

[7] A. Berger and E. Natale, "Assessing the real-world dynamics of dns," in *Proceedings of the 4th international conference on Traffic Monitoring and Analysis*, ser. TMA'12. Berlin, Heidelberg: Springer-Verlag, 2012.

[8] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09. Third International Conference on*, 2009.

[9] C.-S. Chen, S.-S. Tseng, and C.-L. Liu, "A unifying framework for intelligent dns management," *International Journal of Human-Computer Studies*, vol. 58, no. 4, 2003.

[10] D. Wessels, M. Fomenkov, N. Brownlee, and k. claffy, "Measurements and laboratory simulations of the upper dns hierarchy," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science, C. Barakat and I. Pratt, Eds. Springer Berlin Heidelberg, 2004, vol. 3015.

[11] J. S. Otto, M. A. Sánchez, J. P. Rula, and F. E. Bustamante, "Content delivery and the natural evolution of dns: remote dns trends, performance issues and alternative solutions," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, ser. IMC '12. New York, NY, USA: ACM, 2012.

[12] A. J. Yakup Ko and B. Gijsen, "A global reference model of the dns," *Proceedings of DNS EASY 2011 Workshop*, 2012.

[13] J. M. Agosta, J. Chandrashekar, M. Crovella, N. Taft, and D. Ting, "Mixture Models of Endhost Network Traffic," *ArXiv e-prints*, 2012.

[14] N. Brownlee and K. Claffy, "Understanding internet traffic streams: dragonflies and tortoises," *Communications Magazine, IEEE*, vol. 40, no. 10, 2002.

[15] J. Abley and K.Lindqvist, "Operation of Anycast Services," *Internet RFC 4786*, 2006.

[16] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, "Dns performance and the effectiveness of caching," *Networking, IEEE/ACM Transactions on*, vol. 10, no. 5, 2002.

[17] J. Jung, A. Berger, and H. Balakrishnan, "Modeling ttl-based internet caches," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1, 2003.

[18] L. Deri *et al.*, "Unveiling interests and trends using the DNS," in *IADIS Conference on Internet Technologies*, 2012.

[19] F. Fusco and L. Deri, "High speed network traffic analysis with commodity multi-core systems," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010.

[20] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, Nov. 2009.

[21] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW*. New York, NY, USA: Oxford University Press, 2003.

[22] P. Albitz and C. Liu, *DNS and Bind, 4th Edition*, 4th ed. O'Reilly, 2001.

[23] A. Shaikh, R. Tewari, and M. Agrawal, "On the effectiveness of dns-based server selection," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2001.

[24] P. Vixie, "What dns is not," *Queue*, vol. 7, no. 10, 2009.