Exploiting DNS Traffic to Rank Internet Domains

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*—Search engines and Internet service providers rank domains leveraging both on search hits and information gathered through social networks, advertisements and third-party tools such web-browser plugins or mobile applications. Unfortunately, these methods are difficult to implement and deploy as they require substantial a amount of traffic and data to be analysed.**

In this paper we describe a novel methodology for ranking Internet domains based on DNS rather than on HTTP traffic. The advantage is that by analysing a limited amount of traffic it is possible to create comprehensive rankings neither limited to HTTP traffic, nor based on monitoring data provided by Internet users. Although the proposed methodology is general, DNS traffic collected at ".it" ccTLD DNS domain servers has been used to validate this work on a large scale and create daily rankings of Italian Internet domains.

I. Introduction and Related Work

Periodic reports such as Google Zeitgeist [1] and Akamai State of the Internet [2] focus on Internet usage and trends. They contain various types of information such as the number of Internet users, top queries on search engines, popular hashtags on social networks and percentage of spam emails per day. Although the Domain Name System (DNS) can potentially be a good source of data for understanding Internet usage [3], publicly available reports [4] [5] focus only on the number of registered domain per Top Level Domain (TLD), DNS servers performance, or aggregated query reports, without disclosing information about Internet usage and trends. Methods for scoring web pages [6] have been out for years, and are profitably used by search engines to return searches sorted according to the web page score. Similar methods recently appeared also for DNS [7] [8] [9] although to date there are no public DNS traffic reports based on such methods. The authors of this paper have developed a DNS monitoring system [10] [11] able to passively monitor the whole ".it" ccTLD, managed by the Institute of Informatics and Telematics (IIT) of the Italian National Research Council (CNR). As the DNS traffic is one of the core protocols on which the Internet is relying, monitoring DNS activities enables us to analyze relatively little traffic (each .it DNS server receives about 7 million requests/hour) when compared to complex application-protocols probes that instead have to decode a much larger traffic volume — not to mention that they are unable to analyze encrypted traffic. The ".it" DNS monitoring system aims at analyzing DNS traffic in order to understand Internet user trends and interests, and also track anomalous traffic pattern behaviors (e.g. DoS attempts and DNS attacks). Similar to search engines, ranking Internet domains is needed to generate detailed traffic reports focusing on popular domains, and report users about the trends and interests related to `.it` domains. Driven by these motivations, we created novel scoring methodologies for Internet domains, which are based on the DNS traffic passively monitored at the various ".it" DNS servers. The idea is to rank Internet domains exploiting observed DNS queries, in order to create a system able to spot global usage and trends while monitoring relatively little traffic. The primary usage of this ranking is to associate interests to Internet domains. This will enable the creation of a ".it" search engine able to return search results sorted according to the calculated domain rankings. Additional usage of these rankings includes the ability to characterize Internet domains, hence creating new scores based on the domain nature (e.g. sport, business, music), and also identify potential security flaws or DNS misuse.

The main contribution of this paper is a novel and general methodology for ranking Internet domains passively monitoring the DNS protocol. Being us independent from the DNS server implementation, makes this work suitable to monitor a company, and Internet Service Provider (ISP), and also a large ccTLD such as the ".it" ccTLD. Although we have validated our work by monitoring DNS traffic at `.it` authoritative DNS servers, we do not use any peculiarity of the ".it" DNS system, thus making this work pretty general and usable in other contexts.

The rest of the paper is organized as follows. Sect. II describes the motivation behind this work. Sect. III introduces the peculiarities of DNS traffic monitoring, and covers our previous DNS modelling work we used in this paper to rank domains. Sect. IV describes the findings we have obtained when applying our methodology to ".it" DNS traffic monitoring. Finally open issues and future work are described on Sect. V.

II. Motivation

Goal of this work is to monitor DNS traffic not just in terms of number and volume of queries as reported by all monitoring tools, but rather take into account DNS peculiarities in order to rank interests, trends, domains and resolvers. Below we list the main goals we would like to achieve:

1) Define a domain ranking in according to their popularity among resolvers and vice versa.
2) Identify the most popular resolvers so that we can change .it traffic policies with the aim of providing these resolvers a lower response time. This can be achieved for example by minimizing the Round Trip Time (RTT) between the .it name servers and the resolvers using them.
3) Group user's interests. In other words supposing that a resolver queries domainname.it then it is likely that it also queries domainname-1.it, ..domainname-n.it. In the case of web advertisement banners, it can also be used to figure out what are the domain names where domainname.it has placed its advertisement banners just using the DNS traffic. In the case of similar businesses (e.g. domains belonging to the same financial group), it can also be used to discover economical relationships.
4) List resolvers that are likely to misbehave (e.g. do not obey to the Time To Live (TTL) specified for domains they are sending queries for) and that thus need to be monitored more closely as they might perform malicious activities.
5) Rank domains according to the traffic type (e.g. web and email), countries where resolvers are located, density of queries according to the time of the day (i.e. a domain that receives queries according to the Italian working hours is likely to identify a company/individual that is interesting only for domestic users and not a global player).
6) Identify trends in interests and position from an economical standpoint companies that have an Internet presence in the same market segment (e.g. online shopping or web trading). Grouping together similar domains (e.g. e-commerce sites) can be used as a market indicator for speculating how a given market sector performs overtime. Applying the same principle to public persons web sites and political parties, it can be used as litmus test for revealing changes in interests.
7) Identify resolvers that might be used by email spammers, and domains that are likely to be targets of email attacks.

III. Background and Preliminaries

In this section we briefly introduce the DNS system, focusing on the impact that heterogeneous record TTL values have on the number of queries that resolvers issue for Internet domains. Then, we present a conservative approach used to handle such heterogeneities that enabled us to develop graph theoretical models of the DNS. Such models are discussed at the end of the section, after a concise overview of basic graph theoretical definitions.

A. Monitoring DNS Traffic

The domain name system is a hierarchical distributed database organized in a tree of domain names, with the root domain identified by an empty label. Each domain name is served by one or more authoritative name servers. Each DNS tree node contains some resource records which define the information associated with the domain name. The DNS resolver is the client side of the DNS, responsible for performing address resolution by contacting the authoritative name servers for the domain name being resolved. DNS queries can be sent between a resolver and a server, or between two DNS servers. Contrary to other types of traffic such as web or peer-to-peer where end-clients contact the server directly, DNS resolvers perform address resolution on behalf of clients. This means that packets received by an authoritative name server have the resolver IP as source address and there is no reference inside the DNS packet to the original IP that requested the resolution of such name. Furthermore, beside some rare exceptions (e.g. mobile users connected to a cellular network), end-clients are free to use either the resolver provided by the ISP or a public DNS service such as those provided by Google and OpenDNS. The outcome is that estimating the number of clients behind a resolver is not simple at all. This fact together with DNS record caching covered in III-B, makes DNS traffic analysis further challenging.

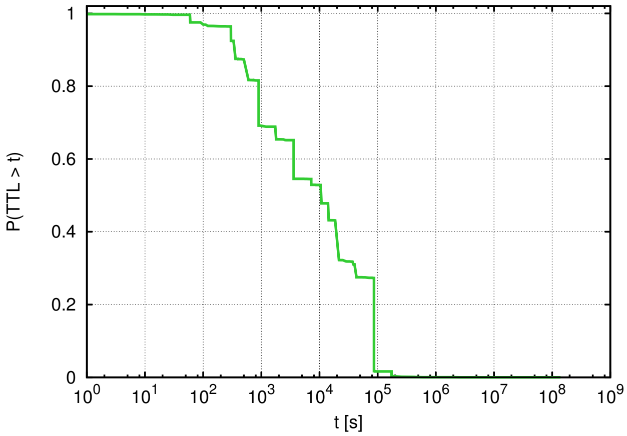

Fig. 1. TTL CCDF for .it Domains

B. Record Caching in the DNS

DNS records have a specified TTL that determines the time for which the given response record can be kept in the resolver cache. The TTL is the mechanism used to reduce the number of queries necessary to resolve a name, as it prevents resolvers from issuing queries for those records that are still in cache. DNS records have non uniform TTL: it can range from 0 (no caching) to days or weeks. These differences in TTL can be observed even within a single domain name, where the domain name server or mail exchange records might have different TTLs. Data caching must be taken into account when monitoring DNS traffic. Indeed, supposing that two domains are equally contacted during the day by a given resolver, the name servers for the domain with lower TTL will receive more queries than the name servers of the other domain, even though both domains have been contacted the same number of times by clients. Figure 1 shows the TTL complementary cumulative distribution function (CCDF) of `.it` domains. As shown in figure, it turn out that over 98% of `.it` domains have a TTL less than 86400 seconds, i.e. 1 day. Hence, we have decided to choose one day as our observation period. Using this approach we are able to model DNS traffic without taking TTL into account: in our graphs the weight on each edge is either zero (no query observed) or one (at least one query was issued during the observation period). While our approach flattens the TTL in order to compare domains with non-uniform TTLs, resolvers must implement data caching according to the TTL. This means that whenever we observe unexpected DNS queries according to the TTL of the queried record, we can safely use this information as a flag for spotting anomalies on DNS traffic that can include misbehaving hosts, probing queries, and malware attacks.

C. Modelling DNS Traffic

We denote with $G = (V, E)$ a *graph* with V being the set of its n nodes and $E \subseteq V \times V$ the set of its edges. Two nodes $i, j \in V$, $i \neq j$, are said to be *adjacent* if the unordered pair (i, j) is in E. A graph is uniquely identified by its n-square *adjacency matrix* $\mathbf{A} = [a_{i,j}]$. Elements $a_{i,j}$ are equal to 1 if nodes $i, j \in V$ $(i, j = 1, \cdots, n)$ are adjacent (i.e. if $(i, j) \in E$) or 0 otherwise — when elements $a_{i,j}$ are allowed to assume real values, then the graph is said to be *weighted*. If V can be divided into two disjoint sets R and D such that each edge links a node in R with a node in D, then G is said to be *bipartite*. In [12] we have defined models of DNS traffic that allows Internet domains, resolvers and their interactions to be represented effectively by means of graphs. Such models are briefly discussed below.

Bipartite Graph Models: We build bipartite graphs by choosing R as the set of resolvers and D as the set of `.it` domains. Connections are obtained placing edges (r, d) between resolvers $r \in R$ and domains $d \in D$, according to different criteria. We have also introduced the following bipartite graphs:

- G_{ALL}: an edge connects r and d iff r issued at least one DNS query for d in the observation period.
- G_{WEB}: an edge connects r and d iff r issued at least one DNS query for d in the observation period for records such as: the domain name with no host specified (e.g. `nic.it`); or the domain name preceded by either `www` or `web`. In essence, we consider only those DNS queries that should be originated uniquely by web traffic, although some queries originated by web traffic might not be taken into account by this method (e.g. `images.domainname.it`).
- G_{MX}: an edge connects r and d iff r issued at least one DNS query for d in the observation period for MX (email) records.

Isolated nodes are removed from graphs, thus resulting in the exclusion of domains (resolvers) not receiving (issuing) at least one query in the observation period.

Common-Neighbours Graph Models: In these models the concept of adjacency between nodes becomes *weighted* with the number of their *common neighbours*. The higher this number, the higher the weight of the edge. In the case of two domains, their common neighbours are those resolvers issuing queries for both of them. In the case of two resolvers, their common neighbours are domains whose names that have been queried by both resolvers in the observation period. Formally, let $N(i, j)$ be the set of neighbours in common between two domains $i, j \in D$ or two resolvers $i, j \in R$ in any of the bipartite graphs above defined. We define as $cn(i, j)$ the number of neighbours they have in common (i.e. $cn(i, j) = |N(i, j)|$), and we take it as the weight of the edge connecting i and j. With this methodology, we obtain two weighted graphs, one for domains and one for resolvers, with node set D and R respectively.

D. Criteria for Ranking Domains and Resolvers

As we have introduced models of the DNS, we can now focus our attention on how to suitably assess the relevance of resolvers and domains. In order to do that, we need to assign them a score that can be used as baseline for a ranking. In particular we define two rankings by sorting resolvers and domains in a decreasing order of *node degree* and *eigenvector centrality*.

Node Degree: The *degree* $d_i = \sum_j a_{i,j}$ of a node i is the number of nodes adjacent with i. Hence, in the case of a domain $d \in D$ in any of the bipartite graphs described above, its degree d_i counts the number of queries resolvers issue for it. Similarly, the degree of a resolver $r \in R$ gives the number of queries it issues for `.it` domains.

Eigenvector Centrality: We choose the relevance of a domain in a way that it is directly proportional to the sum of the relevance of resolvers issuing queries for it. Similarly, the relevance of a resolver is chosen to be directly proportional to the sum of the relevance of domains it issues queries for. Formally, the relevance x_i of resolver (domain) i is measured as $x_i = \lambda^{-1} \sum_{j=1}^{n} a_{i,j} x_j$. This measure can be written in matrix form as the eigenvector equation $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ [13]. In general, there are many eigenvalues for which an eigenvector exists. However, with the additional requirement that components x_i of $\mathbf{x}$ be non-negative, then the Perron-Frobenius theorem ensures that λ is the largest (in absolute value) eigenvalue and $\mathbf{x}$ the corresponding eigenvector. As future work we plan to evaluate additional methods of graph theory for defining new ranks, such as strength, coreness, closeness, and betweeness.

IV. Results and Validation

The `.it` has seven authoritative DNS servers, three of which with anycast addresses. The ".it" DNS monitoring system [11] we used for validating this work, monitors four DNS servers (two anycast and two unicast). Every `.it` DNS server node serves about 40 million requests/day, and we passively collect DNS traffic using a home-grown open-source NetFlow probe [14] featuring a plugin for dissecting DNS

TABLE I
BIPARTITE GRAPHS: TOP .it DEGREE AND EIGENVECTOR CENTRALITY DOMAIN RANKING

	G_{ALL}		G_{WEB}	
Rank	**Degree**	**Eig. Cent.**	**Degree**	**Eig. Cent.**
1.	amazon	corriere	amazon	gazzetta
2.	fastwebnet	rcs	google	corriere
3.	virgilio	gazzetta	corriere	gazzettaobjects
4.	telecomitalia	aruba	excite	corrieredellosport
5.	corriere	virgilio	imdb	softonic
6.	aruba	excite	softonic	tripadvisor
7.	tiscali	gazzettaobjects	gazzetta	vogue
8.	gazzetta	softonic	tripadvisor	agi
9.	rcs	corriereobjects	virgilio	tuttosullavoro
10.	rcsadv	groupon	ebay	virginradioitaly

query/responses. This solution allowed us to be independent from the DNS implementation being used and thus be general enough to use it on different contexts. In this section we present the the monitoring results we observed on Jan 4th, 2013 while monitoring `dns.nic.it`. We omit the results we have obtained on the other three monitoring sites as they are pretty similar to what we will present later on this section. The only differences we observe is that resolvers select an authoritative name server based on its RTT. Hence, for each monitored site the resolvers distribution is different in terms of queries but not in terms of edges, confirming that resolvers randomly select authoritative servers and that they probe servers for lower RTT selection. For goals listed in Sect. II, we use the following approaches.

Goals 1) and 2): We rank domains according to their node degree and eigenvector centrality. Node degree ranks domains in terms of their degree without considering neighbouring resolvers degree. Eigenvector centrality instead takes into account both domains and neighbouring resolvers degree. In Tab. I we compare the results for top `.it` domains when considering all or only web traffic as defined in III-C. Both rankings are similar but not alike. When considering the domain degree we count just the number of resolvers that contacted the domain, without distinguishing across resolvers degree — e.g. a resolver that queried a limited number of domains has the same weight of a resolver that queried many more domains in the same observation period. When using the eigenvector centrality, domains queried by resolvers with higher scores are pushed higher in the ranking. We believe that both ranking criteria are good, but the eigenvector centrality is probably the best as it takes into account neighbouring resolvers degree that give an indication of the size of the population behind such resolver. This is in the assumption that resolvers with higher degree are likely to serve a larger client population than those with a smaller degree.

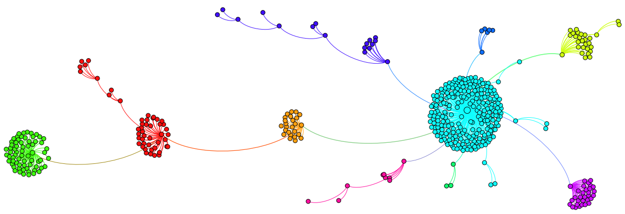

Fig. 2. Common-Neighbours Maximum Spanning Tree for Top 500 `.it` Domains Degree

Goal 3): We build the common-neighbours graph of `.it` Internet domains. In Fig. 2 we have selected the top 500 domains according to their eigenvector centrality score from G_{ALL} and created a their common-neighbours graph as described in Sect. III. Then, we extracted the maximum spanning tree [15] from the latter graph. If domains d_1 and d_2 are connected in the minimum spanning tree, there is no domain d_3 such that: *i)* d_1 has a number of common neighbours with d_3 greater than the number it has with d_2; and *ii)* d_2 has a number of common neighbours with d_3 greater than the number it has with d_3. Formally, $cn(d_1, d_2) > cn(d_1, d_3) + cn(d_2, d_3)$.

Goal 4): We use a combined approach.

- The modelling methodology defined in [12] takes TTL into account. For each tuple `<resolver, queried domain, TTL query response>` we should not observe at each monitoring point a query more frequent than the TTL specified. If this property is not respected, then the resolver is likely to use a faulty software or to be a scanner. In both cases it is worth to be analyzed more in depth.
- For each resolver we keep the ratio of positive and negative replies, and we group it on the autonomous system (AS) such resolver belongs to. This is in order to also take into account other resolvers (e.g. secondary DNSs) belonging to the same administrative domain. If the ratio exceeds a certain threshold we mark this activity as suspicious. In fact, in case of negative DNS replies (e.g. NXDOMAIN), the resolver must also cache these responses and avoid repeating negative queries similar to what happens with positive replies. Furthermore high negative responses ratio, often identify scanners attempting to guess the registered domain names, given that such list is not publicly available.

Goal 5): Tab. I shows different types of ranking based on the nature of traffic. As previously explained, data caching in DNS does not prevent us from analysing data at a granularity lower than a day, and thus just compute a daily ranking. Nevertheless, this does not prevent us from periodically accounting the number of observed domain queries. This has enabled us to:

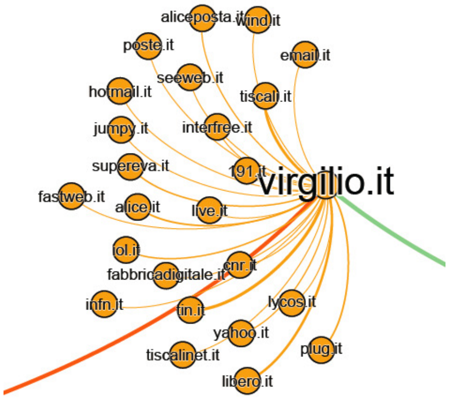

Fig. 3. Common-Neighbours Maximum Spanning Tree for `virgilio.it`

- Highlight periodicity in traffic, such as identify domains that are mostly accessed during the day including webmail portals and (many but not all) news sites written in Italian.
- Spot hosts used for activities that happen during the whole day such as torrent tracker sites.
- Highlight hosts that receive very periodic contacts from specific resolvers, that might be due to remote monitoring activities and that are also trapped by TTL algorithm used for 4).

Goal 6): we have used the common-neighbours graph also used for 3) to automatically cluster domains. In Fig. 3 we zoomed a region of Fig. 2 to spot the links of a large Italian content provider. Although our approach is based uniquely on the domains degree with no knowledge of the type of information hosted by domains web sites, the spanning tree algorithm has been able to place on the same cluster additional domains of domestic ISPs and Internet content providers. The same behaviour can be found on the additional clusters of the Fig. 2 that have not been included due to space constraints. According to our knowledge, domain clustering happens when:

- Domains have economical relationships. For instance domains such as `fiat.it`, `alfaromeo.it` and `abarth.it` belong to the same cluster as their web sites contain cross links to all these domains that belong to the Fiat group.
- Domains are similar in content as shown in Fig. 3.
- Domains have some ”side relationships”. For instance `amazon.it` has several edges in common with peer-to-peer and torrent tracker sites. We believe that such sites use amazon to either show multimedia artwork of shared files, or perhaps people first search on amazon the stuff they are interested in, and the access such sites for (illegally) downloading it.

Goal 3): We use the G_{MX} bipartite graph described in III-C for focusing on email traffic. Currently, we are able to use this information for emitting alerts only if the daily degree of domains change suddenly with respect to previous days. As future work we plan to characterize domains, and thus create a more advance alerting system. For instance domains of ISPs or large institutions can have a higher alert threshold than domains of smaller institutions. The ratio between G_{MX} and G_{ALL} can also be used to spot sites that mostly perform mail activity, and also that might be worth to further investigation.

In summary, the DNS traffic model we have defined has enabled us to reach our project goals. As stated in III-D, we are currently evaluating additional methods for assigning scores to domains and resolvers, in order to create additional rankings.

V. Future Work Items

The described methodology not only allowed us to rank Internet domains, but also enabled us to identify those resolvers that do not honour the TTL and thus that violate the principles of DNS. In average 5% of resolvers fall into this category. A future work item is to refine our methodology in order to insulate malicious activities from misconfigured or faulty DNS resolvers and thus generate alerts to the domain administrators.

Large web sites host images and media on hosts other than www (e.g. `images.domainname.it`). For some sites, our methodology reports higher ranking for media sites with respect to the corresponding www site. Our feeling is that people reference media on social networks such as FaceBook, Twitter, and Pinterest thus increasing the ranking of these sites. Another explanation of this fact could be that banners are often hosted on these media sites, thus increasing their ranking. In the coming months, we plan to analyse this fact in detail in order to figure our more information about the causes that originate it.

VI. Conclusion

This paper has described novel methodologies for ranking Internet domains using DNS traffic. The main advantage of our approaches is that the monitored traffic to be used for creating rankings is very limited with respect to other protocols such as HTTP or social networks analysis. The validation phase has demonstrated that using the proposed methodologies enable, among other things, to: successfully rank resolvers and Internet domains according to different criteria; automatically cluster domains containing similar information and interests; and discover malicious activities using the DNS traffic otherwise difficult to identify by other means.

References

[1] “Google Zeitgeist 2012,” Google Inc, Tech. Rep., 2012.

[2] “State of the Internet: Q4 2012 Report,” Akamai Technologies, Tech. Rep. Q4-2012, 2012.

[3] C. Huang, D. Maltz, J. Li, and A. Greenberg, “Public dns system and global traffic management,” in *INFOCOM, 2011 Proceedings IEEE*, april 2011, pp. 2615 –2623.

[4] “State of the Domain,” Verisign Inc, Tech. Rep., 2012.

[5] “OpenDNS stats,” OpenDNS Inc, Tech. Rep., 2013.

[6] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” in *Seventh International World-Wide Web Conference (WWW 1998)*, 1998. [Online]. Available: http://ilpubs.stanford.edu:8090/361/

[7] A. Holmes *et al.*, “Domain traffic ranking,” European Patent 2012/EP241 753, February 15, 2012.

[8] A. T. Sullivan, “Methods and systems for node ranking based on dns session data,” US Patent 2012/8 090 726, January 3, 2012.

[9] A. Holmes *et al.*, “Existent domain name DNS traffic capture and analysis,” US Patent 2010/0 257 266, October 14, 2010.

[10] L. Deri, L. Trombacchi, M. Martinelli, and D. Vannozzi, “A distributed dns traffic monitoring system,” in *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, aug. 2012, pp. 30 –35.

[11] L. Deri *et al.*, “Unveiling interests and trends using the DNS,” in *IADIS Conference on Internet Technologies*, 2012. [Online]. Available: http://www.its-conf.org

[12] L. Deri, S. Mainardi, M. Martinelli, and E. Gregori, “Graph theoretical models of dns traffic,” January 2013.

[13] P. Bonacich, “Power and Centrality: A Family of Measures,” *American Journal of Sociology*, vol. 92, no. 5, 1987.

[14] 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, pp. 218–224. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879169

[15] A. P. Punnen, “A linear time algorithm for the maximum capacity path problem,” *European Journal of Operational Research*, vol. 53, no. 3, 1991.