Public DNS Resolvers: Friends or Foes?

(Technical Report - August 2013)

Alessandro Finamore, Ignacio Bermudez, Marco Mellia Politecnico di Torino Email: lastname@tlc.polito.it

ABSTRACT

Public DNS resolvers are offered to Internet users with the promise to improve web browsing experience. However, it has been shown that their usage can impair performance when accessing resources hosted by CDNs. In fact, public resolvers could redirect users to not-optimal CDN replica servers from which users experience lower performance.

Differently from previous works based on active benchmarks, we present an extensive study of the implications of DNS resolver choice for customers of a large European ISP. Our results show how DNS prefetching and caching policies adopted by web browsers mask (if present) the benefits of faster DNS resolution: in practice, benefits are present for less than 3% of requests. We then precisely quantify the actual performance that customers using different resolvers obtain when browsing the web. We show, for the first time to the best of our knowledge, that the usage of public DNS resolvers causes sensible impairments even when advanced technologies such as the one offered by ECS are in place. For instance, the median of the download throughput decreases by 35%, with some services loosing even more than 50%.

This suggests that, for the customers of the ISP under analysis, public DNS resolvers do not represent a worth alternative. Technical solutions like the one offered by ECS are not yet mature to mitigate the problems. This calls for further improvements of the synergies between DNS resolvers and CDN operators.

1. INTRODUCTION AND MOTIVATION

Nowadays, with the explosion of Content Delivery Networks (CDNs) and Cloud based services, popular content can be fetched by multiple servers widely distributed all over the world [1, 2]. In this scenario, the Domain Name System (DNS) is used by CDNs to map the requested resource, i.e., a hostname, to the "best" server.

Internet Service Providers (ISPs) are accustomed to offer DNS resolvers to their customers. Since these resolvers are located in the same ISP network, they are referred to as Local DNS resolvers (LDNS). During a DNS resolution, the end-user client contacts the configured LDNS. This in turn contacts the CDN authoritative domain name server related to the requested hostname. In this process the LDNS represents all ISP customers using it, and allows the CDN to take a wise decision during the mapping operation. In fact, CDN resolvers can easily identify the requests as coming from a particular ISP. Based on this information (and other metrics related to CDN load balancing and traffic demand), they can map the requested hostname to the "best" CDN server [3,

4], i.e., a server typically close to the end-users, possibly within the same ISP network originating the request.

Recently, Internet users are offered the freedom to configure so called public DNS resolvers, i.e., third-party DNS resolvers such as Google DNS and OpenDNS. These resolvers are becoming customary among users [4] with the popular belief that they perform better than the LDNS resolvers. Conversely, several studies in the literature have shown possible negative impacts of the adoption of public DNS resolvers when accessing CDNs content [3, 4, 5, 6]. In fact, given that public resolvers correspond to few tens of servers coarsely scattered worldwide [3], the approximation of the end-user ISP with such servers results coarse, and the mapping operations too generic. For comparison, the Verizon DNS infrastructure partition U.S. customers among 60 DNS resolvers¹, thus it opens to the CDN resolvers a much more fine grained information.

The negative interplay between DNS and CDNs when using public DNS resolvers has been already highlighted in the literature. A few solutions have been thus proposed to mitigate problems [7, 8, 9]. In particular, the Global Internet Speedup consortium², which includes Google, OpenDNS and EdgeCast, has proposed a DNS extension called Extended-DNS Client Subnet (ECS) [9]. Potentially, ECS has the ability to expose the originating client IP address to the CDN nameserver, thus re-enabling accurate mapping.

Despite all these efforts, the actual impact of the DNS resolver choice on the Internet users' web browsing experience has only been marginally discussed in the literature. This is because, to the best of our knowledge, all previous works have adopted active methodologies that consider a handful of benchmarking objects. This clearly expose the problem, but lack of generalization when considering the actual quantification of browsing performance. In this paper, we fill this gap by exploiting a very large dataset of passive measurements collected in the wild from the network of a major European ISP. More than 35,000 ADSL home customers from three different cities have been monitored for 44 days. We use this dataset to factorize the components that affect web browsing performance, including i) DNS resolution time, and, more interesting, ii) the actual object download time and throughput obtained by users browsing the web.

Considering DNS resolution time, in contrast to [10], we show that modern web browsers and operating systems heavily adopt DNS prefetching and caching policies that makes DNS resolution time not critical. To assess the actual brows-

¹Personal conversation with Paul Mockapetris.

²http://www.afasterinternet.com/

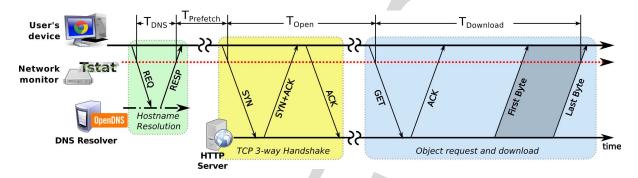


Figure 1: Typical evolution of an HTTP object download and timing measurements definition.

ing performance, we focus on popular services that could be affected by wrong CDN replica server selection, namely content download from Apple iTunes/App Store (hosted by Akamai), YouTube videos and EdgeCast CDN services. Our findings show that customers using public DNS resolvers consistently present worse performance than those using LDNS. For instance, the median of download rate decreases from 2 Mb/s to 1.34 Mb/s (-33%) and 1.64 Mb/s (-18%) for OpenDNS and Google DNS adopters, respectively. Finally, we consider the EdgeCast use-case, one of the first (and popular) CDNs to have rolled out the novel ECS solution. Also in this case, measurements highlight the presence of significant impairments with respect to LDNS. This is possibly due to a still limited optimization of the ECS parameters and internal algorithms.

Our conclusion is that, for web browsing performance, there is no good motivation to opt for a public DNS resolver. Some users might still prefer to opt for public resolvers because of specific features such as parental control, malware protection, etc. In such cases, users should be warned of the possible impairments. More importantly, CDN resolvers should further optimize their mapping mechanisms when ECS is in place.

2. RELATED WORK

Public DNS resolvers performance have recently been subject of several studies [3, 4, 5, 6, 11]. These works capture the interplay between DNS and CDNs by running active experiments on instrumented clients using ad-hoc scripting [5], software plugins [4, 6], or javascript code run by browsers when accessing a popular page [3]. Results highlight that local resolvers are typically closer to end-users [3, 11], with public resolvers being scarcely deployed [4, 6]. As such, public resolvers have negative impact on CDN mapping policies [5].

Since all previous works embrace active benchmarks, they are valid works to pinpoint eventual issues, but they fail to provide any quantitative study of how large the problem could be. This is particularly true when considering the actual end-user performance. For instance, in [3, 4, 6] a *single* web object is considered when investigating web browsing performance. Given the very heterogeneous and dynamic nature of the web and of CDNs policies [12, 13], this cannot be considered representative of any actual estimation of the problem.

In this work, we corroborate and complete the previous works by using passive measurements. This allows us to better quantify actual performance of popular CDN services, and to highlight components that are hardly observed using active experiments. First, we discuss the impact of client DNS prefetching and caching on resolver choice. Our results are surprisingly different from the one in [10] (60% of DNS requests results cached in our study versus only 13% in [10]). This is possibly due to the number of considered end-users (more than 35,000 households versus the 90). Second, we measure and precisely quantify web objects download time and throughput using a humongous number of actual objects. Finally, we study the stability over time of measurements using more than one month of data.

The evidence of the possible negative interplay among public resolver and CDN mapping algorithms spawn some research effort to provide technical solutions to mitigate the problem [7, 8, 9]. In particular, ECS [9], proposed by the the Global Internet Speedup consortium², is attracting a lot of attention. The idea is to expose in the DNS queries operated by recursive resolvers a truncated version of the original end-user IP address (e.g., using a /24 mask). This solution is already adopted among the partners of the consortium. However, other CDNs, e.g., Akamai, refrain to adopt it since it further complicates the mapping policies. These are in fact optimized to work with a limited set of possible "sources", i.e., LDNS resolvers, that currently aggregate all clients behind them [13].

If on the one hand ECS should guarantee performance improvement, on the other hand it is not easy to quantify up to which extent. In [4, 6] authors used active experiments to show that benefits were visible for public DNS resolvers when using the ECS extension. However, a recent work demonstrates that ECS enabled resolvers are sensible to the variation of the truncation of the client IP address [14], rising the question on how to properly tune this parameter. Again, all previous works adopt active experiments. With passive measurements, we contribute to the discussion about the ECS benefits.

3. METHODOLOGY

We build upon Tstat [15] and DN-Hunter [16]. Tstat is a passive sniffer that rebuilds TCP connections in real time and reports text log files having more than 100 different metrics for each connection. DN-Hunter is another passive methodology capable to link DNS messages to the TCP connection they are related to. More in details, given a DNS request/response exchange, DN-Hunter extracts client and DNS resolver IP addresses, the requested hostname, the list

Table 1: Rules used to identify traffic classes

Classname	MaxMind	Regex
Facebook Dynamic	facebook	none
Facebook Static	akamai*	FQDN does have facebook or fqdn
Akamai-Other	akamai*	opposite as Facebook Static
EdgeCast	edgecast	none
YouTube	none	.*lscache[0-9]+\.*youtube\.com
		.*lscache.*youtube.com
		$r[0-9]+\-\-*\$.youtube\.com
Apple iTunes/App	none	.*\.phobos[]apple\.com
		$.*\.$ phobos[]apple $\.$ com $*$

^{* =} we added also a list of other names corresponding to caches inside ISPs which we cannot report due to NdA.

of returned server IP addresses, and the timestamps to compute the query response time. When later a TCP connection is opened, DN-Hunter retrieves the associated DNS data and includes them in the set of logged TCP metrics. For more information we direct the readers to [16]. Finally, servers are associated to the organization which own them by mean of the MaxMind Organization Database³. This allows us to easily identify traffic related to specific CDNs.

We instrumented three Points of Presence (PoPs) of a large European ISP to collect statistics from about 35,000 ADSL residential customers⁴. In this work, we leverage a 44 day long dataset, from March 18th, 2013. Overall, the dataset includes 1.2 billions HTTP transactions.

Fig. 1 sketches the typical pattern followed by a user downloading content from a web server using HTTP. From top to bottom, four time lines are reported, referring to the user's device, the Tstat sniffer, the DNS resolver and the HTTP server, respectively. From left to right, we highlight three phases: (i) the client contacts the DNS resolver to obtain the IP address of the web server; (ii) the TCP connection is opened towards the web server; (iii) the content is requested and received by the client.

To capture these dynamics, Tstat records several timestamps: T_{DNS} captures the DNS resolver response time. $T_{Prefetch}$ corresponds to the time between the DNS response and the TCP SYN. T_{Open} is the time between the start of the TCP 3-way handshake and the packet carrying the first HTTP request. Notice that some browsers (e.g., Chrome) can inflate these two latency using DNS prefetch and TCP preconnect respectively (see Sec.4.2). At last, $T_{Download}$ measures the elapsed time when downloading the actual content. For connections having more than 500 kB⁵ (typically videos, ipa/apk mobile apps, software updates, etc.) we report also the downloaded bytes and $T_{Download}$.

In this work, we focus on specific CDN as aggregate and on specific services they offer. As said previously, we use the MaxMind Organization name to identify CDNs while we apply some regular expression to the Fully Qualified Domain Name (FQDN) of the HTTP requests to stop specific services. Table 1 report the list of rules used to identify traffic classes. Notice that some Akamai caches can be hosted inside ISPs network so the name which is also reflected by

Table 2: Breakdown of the top 20 DNS resolvers over one week of data.

				Flo	WS	Us	ers
	#	Name	IP	no.[M]	%	no.	%
١	1	ISP	LDNS1	173.703	61.74	28169	92.23
	2	ISP	LDNS2	59.882	21.28	26050	85.29
	3	googleA	8.8.8.8	25.988	9.24	4866	15.93
		googleB	8.8.4.4	4.871	1.73	3720	12.18
	5	ISP	LDNS3	2.878	1.02	329	1.08
	6	opendns1	208.67.222.222	2.833	1.01	678	2.22
	7	ISP	LDNS4	1.308	0.46	104	0.34
	8	ISP	LDNS5	1.304	0.46	176	0.58
	9	rapidns1	176.31.229.24	1.191	0.42	430	1.41
	10	opendns2	208.67.220.220	1.177	0.42	603	1.97
		rapidns2	176.31.229.25	0.594	0.21	399	1.31
		ISP	LDNS6	0.588	0.21	186	0.61
	13	ISP	LDNS7	0.542	0.19	123	0.40
		unknown	156.154.70.22	0.461	0.16	154	0.50
		recursive.dns1		0.441	0.16	162	0.53
	_	ISP-other	LDNS1	0.357	0.13	69	0.23
-		ISP	LDNS8	0.339	0.12	68	0.22
		ISP-other	LDNS2	0.125	0.04	32	0.10
		kpnqwest1	212.97.32.2	0.107	0.04	10	0.03
	20	114dns	114.114.114.114	0.101	0.04	11	0.04
		-	others	2.570	0.91	-	-
			total	281.359	100	35543	100

ISP = LDNS resolvers of the considered ISP;

ISP-other = LDNS resolvers of another ISP (misconf. clients);

Google DNS = (8.8.8.8, 8.8.4.4);

OpenDNS = (208.67.222.222, 208.67.222.220);

RapiDNS = (176.31.229.24, 176.31.229.25)

the name returned by the MaxMind database, i.e., simply selecting all flows having akamai in the organization name does not allow to identify all the Akamai traffic.

4. DNS RESOLVER POPULARITY AND PERFORMANCE

4.1 Resolvers Popularity

Table 2 details the ranking of the top 20 resolvers with respect to the number of adopters, showing also the number of flows for each resolvers. Results refer to the first week of data. As already found in literature, LDNS resolvers are the most popular among ISP customers. Notice that ISP customers requests are handled using a few different servers. Overall LDNS of the considered ISP are contacted by more than 95% of the users over the week.

Google DNS resolvers are the second choice. Interestingly, Google DNS A and B have a similar share of customers but a different a different amount of requests. These resolvers account for about 16% of customers. OpenDNS is used by more than 900 households, a share of 2.6%.

Notice that users' terminals can be configured to use multiple DNS resolvers, e.g., LDNS and Google DNS resolvers can be used at the same time. More specifically, we found only 7% of customers contacting a single DNS resolver, and 80% contacting 2 resolvers. However, only less than 2% of customers are using resolvers of different organization (e.g., both LDNS and Google DNS) during the week.

Overall, we found more than 40 DNS resolvers. These include less popular resolvers (e.g., RapiDNS, UltraDNS, etc.), misconfigured terminals (e.g., as specified in the table, users trying to contact DNS resolvers of a different ISP that

³http://www.maxmind.com/en/organization

⁴The ISP assigns static IP addresses to customers which simplify the continuous traffic analysis.

 $^{^5 {\}rm https://developers.google.com/speed/articles/web-metrics}$

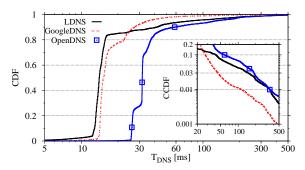


Figure 2: T_{DNS} CDF for the first week of data.

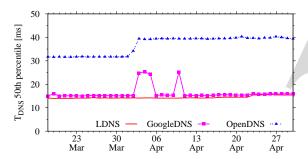


Figure 3: Evolution of the T_{DNS} 50th percentiles.

would refuse to respond) and a few cases of DNS hijacking due to some malware. They account for about 5% of the cases, but individually they corresponds to few terminals, making them statistically not significant. As such, in the following we focus our attention only on LDNS, Google DNS and OpenDNS. We also group together the different anycast IP addresses (e.g., 8.8.8.8 and 8.8.4.4), as well the different LDNS resolvers, since we found marginal differences when considered individually.

4.2 Resolvers Performance

We start our analysis by focusing on the resolver response time. Fig. 2 reports the T_{DNS} Cumulative Distribution Function (CDF) for the first 7 days of the dataset. The steep vertical shape of the distributions reflects the physical distance between the resolver and the client. In our case, LDNS offers the fastest resolution time. In particular, the 80th percentile of the distribution is 15 ms, 22 ms and 37 ms for LDNS, Google DNS and OpenDNS respectively. For other ISPs, these measurements will be different, but LDNS would be the closest with very high probability [3, 6, 4]. Interestingly, OpenDNS distribution presents knees suggesting the presence of multiple servers/paths (we verified that the steps correspond to resolvers located in different locations). This is less evident for Google DNS, with only a small bump at 20ms.

In case the resolver incurs in recursion, the latency of the resolution increases (highlighted in the inset with the distribution tails). In this case, Google DNS resolvers are the fastest with only 1.8% of requests requiring more than 60 ms. LDNS comes second, with only 6.3% of $T_{DNS} > 60$ ms. Overall, Google DNS latency improvements are significant for less than 5% of queries. OpenDNS performs the worst, but also in this case only 8.3% of request suffer $T_{DNS} > 60$ ms. In summary, fastest resolution time claimed by public resolvers is quite marginal (if present).

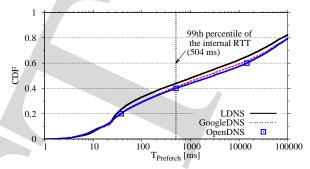


Figure 4: $T_{Prefetch}$ CDF. Considering the 99th percentile of the internal RTT distribution as a conservative threshold, caching and prefetching affect about 60% of requests.

To capture the stability of performance over time, Fig. 3 shows the evolution of the 50th percentile of T_{DNS} distributions computed on a daily base on the entire dataset. We can notice that curves are in general flats (same results are found considering smaller time scales and other percentiles). Surprisingly, Google DNS and OpenDNS present significant and sudden shifts of about 10 ms. The same effect has been observed considering all vantage points (results not reported here for brevity). Such sudden jumps are possibly due to changes in the routing plane, but unfortunately the collected data do not allow us to further investigate on the subject. Overall, robustness and stability of performance are better for LDNS.

4.3 Impact of DNS Prefetching and Caching

Despite the quantification of the resolving latency, a more fundamental question is: "Does the DNS resolution time really matter?". Modern browsers are in fact known to adopt aggressive DNS prefetching [16] and caching. When the user accesses a web page, the browser resolves all hostnames of resources needed to render the web page. It might also resolve hostnames of resources needed later, i.e., it prefetchs the DNS resolution [10]. In such cases, the DNS resolution has less impact on the overall web browsing latencies.

To quantify DNS prefetching implications, Fig. 4 reports the $T_{Prefetch}$ CDF. Intuitively, a long $T_{Prefetch}$ occurs when the hostname has been cached or prefetched by the user's terminal. Since collected measurements reflect the timing at the passive probe, the time between the DNS response and the TCP SYN packets is also related to the Round Trip Time between the probe and users' devices (internal RTT). In our case, the 99th percentile corresponds to 504 ms (highlighted by a vertical bar in Fig. 4). Using this value as a conservative threshold, we can see that about 60% of HTTP connections enjoyed prefetching and caching. This result is in contrast with the analysis reported in [10], where only 13% of HTTP transactions were estimated having DNS prefetching. However, in [10] only 90 households have been considered.

Correlating this with the fact that DNS latency improvements are significant for less than 5% of queries, the eventual benefit of using public resolvers is effective for less than 3% of HTTP transactions. We leave the reader to conclude if this is worth.

⁶http://www.igvita.com/2012/06/04/ chrome-networking-dns-prefetch-and-tcp-preconnect/

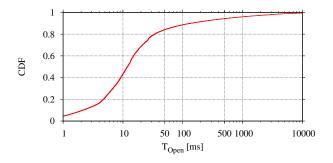


Figure 5: T_{Open} CDF for 1 week of data.

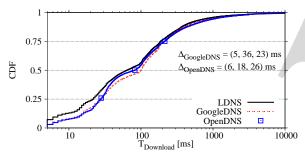


Figure 6: $T_{Download}$ CDF for the whole data set. The labels report the Δ indexes of the 25th, 50th, and 75th percentiles of the distributions.

5. WEB BROWSING PERFORMANCE

We now focus on the implication of DNS mapping policies on end-users web navigation. As shown in Fig. 1, we consider the object download time $T_{Download}$ which excludes the TCP three way handshake duration, T_{Open} . Indeed, modern web browsers can adopt TCP preconnection⁶, i.e., they start opening TCP connections before actually having a HTTP request to be transmitted.

In our dataset we have observed that TCP preconnect is not very common. Fig. 5 details the T_{Open} CDF. As we can see, less than 10% of HTTP connections have $T_{Open} > 500$ ms. Yet, to make a conservative choice, in the remaining of the paper we focus only on $T_{Download}$ and the related download rate.

To easy the comparison, we report always the distance between the quartiles of different CDFs. This is particularly instrumental when tracking differences over time. More in details, we consider the LDNS CDF as reference, and compute the difference between the $X \in \{25th, 50th, 75th\}$ percentile of the public resolver CDFs and the LDNS CDF. We call such differences Δ . When considering $T_{Download}$, positives values of Δ indicate that public resolver users experience extra latency during a download. Conversely, negative values of Δ when considering the download rate indicate that public resolver users experience a throughput loss.

5.1 Overall Impact

Fig. 6 shows the $T_{Download}$ CDF considering all HTTP connections in the whole dataset. Distributions present small absolute differences, with LDNS offering the best performance (note the logscale on the x-axis). The Δ indexes are all positive. For instance, consider Δ for the 50th percentile: OpenDNS causes 18 ms (+28%) of extra delay; Google DNS causes an extra latency of 36 ms (+56%).

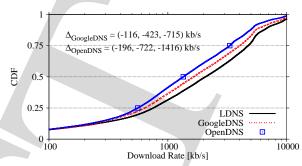


Figure 7: Download throughput CDF for the whole dataset.

More surprisingly, the download rate distributions reported in Fig. 7 present sharper differences (again, note the x-axis logscale). LDNS offers the best performance with both Google DNS and OpenDNS showing significant performance losses. For example, Δ for the 75th percentile of OpenDNS and Google DNS are -1.4 Mb/s (-30%) and -715 kb/s (-15%), respectively.

The differences are even more critical when considering specific services. For instance, consider the download of content from Apple iTunes and Apps Store services which is used regularly by more than 30% of monitored customers. Apple leverages the Akamai CDN to distribute content. Thus, this is a benchmark for Akamai CDN too. To the best of our knowledge, Akamai does not support ECS yet.

Fig. 8(top-left) shows the download rate CDF, along with the Δ indexes. As before, LDNS offers the best performance. Astonishingly, both Google DNS and OpenDNS users clearly suffer of significant performance impairments: the median value of the throughput loss is -2.5 Mb/s (-57%) and -1.2 Mb/s (-26%) respectively. These results confirm the difficulties of CDNs DNS mapping when public resolvers are adopted. Notice that this is an important contribution since such sharp differences have never been captured in previous studies.

5.2 Performance Evolution

A natural question is to verify if these results are consistent over time, and over different services. For instance, what if we consider CDNs that already support ECS extension? Indeed, both Google DNS and OpenDNS support ECS (LDNS of the considered ISP does not support it). We continue the analysis on Apple iTunes/App previously introduced and then the we focus on YouTube video download and EdgeCast CDN as use-cases. Since YouTube is a Google service, one would expect to obtain performance improvements when adopting Google DNS. Both indeed are managed by the same organization. We selected EdgeCast instead because is one popular CDNs that supports ECS. We further report the analysis of Facebook static (served by the Akamai CDN) and dynamic content (served by Facebook servers), and other Akamai CDN content.

Apple iTunes/App: Fig. 8 shows the metrics related to Apple iTunes/App download. The first row details results for the download rate. On the left is reported the overall CDF, previously discussed, while on the right there is the Δ indexes. Overall performance are worse for public resolvers but occasionally they perform better. We believe this is related to the fact that, even if the service is popular, on a

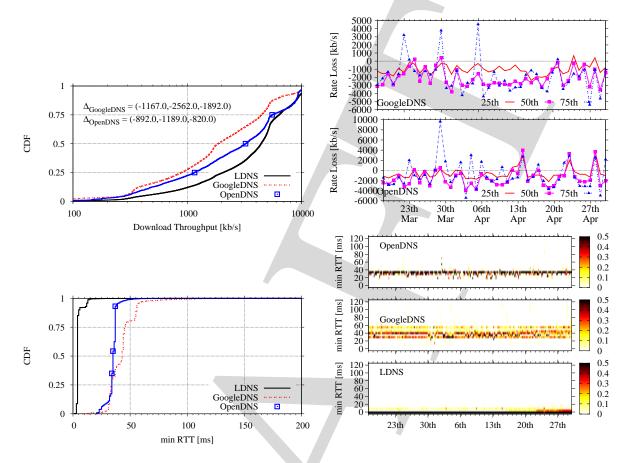


Figure 8: Apple iTunes/App metrics

daily base the number of customers using it might be too small and introduce some bias in the measure.

The second row of Fig. 8 shows the evolution on the minimum RTT. Used as a metric of distance, it allows to study how the resolvers map the request with respect to users location. The plot of the left show the overall CDF. We can notice that, while LDNS maps users to servers at about 5 ms, both public resolvers redirect requests further away.

The plot of the right further details the evolution on the mapping with an heatmap. For each group of 4 consecutive hours, we created the empirical distribution of the min RTT using a bin size of 2 ms. These distributions are in turn mapped to a colorscale where the darker the color, the higher is the value of the distribution. The heatmap show flat trends which is consistent with the CDF. In particular, notice how the steps of the Google DNS CDF corresponds to different horizontal lines in the heatmap.

YouTube Video Streaming: Fig. 9-1st-Row reports all the metrics extracted for YouTube video streaming. As before, the figure on the left shows the download rate CDF considering the whole dataset, while on the right we report the evolution of the Δ indexes over time. We observe consistently significant negative values for Δ , indicating that LDNS guarantees the best performance. Only on the 24th of March, and only for OpenDNS users, download rate was better than the one enjoyed by LDNS users. We believe this is due to the representativeness of the OpenDNS samples on that day (a Sunday with only about 50 users being ac-

tive during peak hours while for other weekends we found hundreds of users being active).

Despite the adoption of ECS and being managed by the same organization, customers using Google DNS suffer severe performance losses, with Δ index of 50th percentile being in the order of -30% (-800 kb/s), a quite significant impairment. OpenDNS performs a little worse.

Considering the evolution of the minimum RTT detailed in Fig. 9-2nd-Row a couple of observation holds. First, the overall minimum RTT CDF of LDNS and Google DNS (left plot) presents marginal differences with about 73% of request being served from servers located at a distane lower than 5 ms. This corresponds to only 30% of requests for OpenDNS, which distribution presents also a longer tail. Second, OpenDNS presents a more variable mapping over time

EdgeCast Content: We conclude the analysis studying the performance when accessing EdgeCast content. This CDN hosts popular services such as WordPress, Tumblr, AddThis and popular adult streaming service. We found 75% of users accessing EdgeCast services in the dataset. Rather than pinpointing differences on specific services, we are interested in identifying macroscopic effects so we study the EdgeCast traffic as aggregate.

We focus on $T_{Download}$ for short flows. As before, for each day, we compute the Δ indexes among CDFs. Fig. 10 shows the result. We can see quite consistent flat trends over time: Google DNS (top plot) redirects users to servers which add

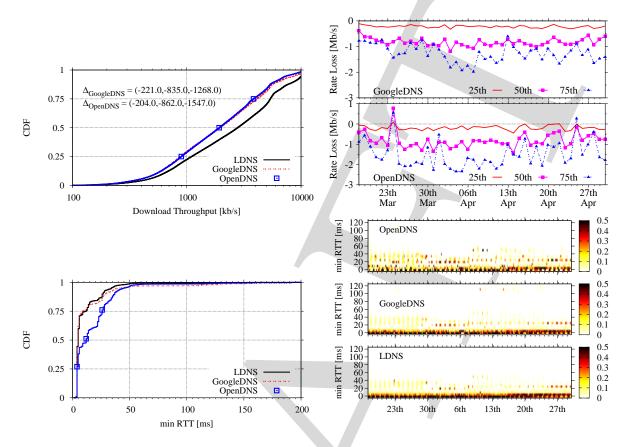


Figure 9: YouTube video streaming metrics

about +100 ms of extra latency (notice that 25th, 50th and 75th percentile show all the same offset). Interestingly, Δ drops close to zero for a few days. OpenDNS (bottom plot) instead presents performance comparable to LDNS with differences only for the 75th percentile, i.e., in the distribution tail.

Since we are focusing on small objects, the download performance are clearly related to "distance" between users device and the contacted CDN server. We consider the minimum RTT experienced by the TCP connections as a measure of such distance. Fig. 10-3rd-row shows its evolution as heatmap. For each group of 4 consecutive hours, we compute the distribution of the minimum RTT using 2 ms bins. Each sample is mapped to a color scale. The higher is the value, the darker is the color. Consider the LDNS (bottom plot). We can see that the majority of traffic is directed to CDN servers having RTT lower than 20 ms (definitively within Europe), Google DNS (middle plot) directs most users to servers with RTT higher than 100 ms (located in the U.S.). Changes are visible over time, corresponding to different mapping that results in changes in the CDN policies. This can affect end-users' performance. For instance, drops in the RTT corresponds to the drops of Δ indexes in Fig. 10.

OpenDNS presents a server mapping similar to the LDNS one. However, as highlighted by the alternating darker colors, the mapping that EdgeCast policies apply for OpenDNS is more variable. This justify the extra latency indicated by the Δ indexes of the 75th percentile in Fig. 10.

A curios effect can be noticed around April 14th. Fig. 10 shows a change of server mapping for both LDNS and OpenDNS while no sensible variations are visible for Google DNS. However, such variations are not enough to justify the +150 ms extra latency experienced by LDNS users when compared to OpenDNS users. We believe than that between April 13th and 14th a problem along the network path or internally to the CDN has created some impairments to LDNS users. We can in fact notice that for LDNS some changes are rolled out on April 13th as well, a possible symptom of some issues.

At the high level, results show that, even when ECS is enabled, significant differences are possible in the mapping policies. This in turn causes impairments. We believe this is due to yet non-optimal tuning of the system, indicating that the ECS deployment is not mature.

Plots in the middle of Fig. 10 show results related to down-load rate. Again, results are better for LDNS with the media value losing -529 kb/s (-30%) and 613 kb/s (34%) for Google DNS and OpenDNS respectively.

Facebook Content: Facebook content has to be divided in two categories: dynamic and static. The former is served by Facebook data centers while the latter is hosted on the Akamai CDN. Let us consider the dynamic component first. Fig. 11 collects all the metrics analyzed. We can notice that $T_{Download}$ presents marginal differences in the body of the distributions while these can be large on the tails.

The minimum RTT show the usage of data centers located in Europe (20 ms and 30 ms) and U.S. (>100 ms). Interest-

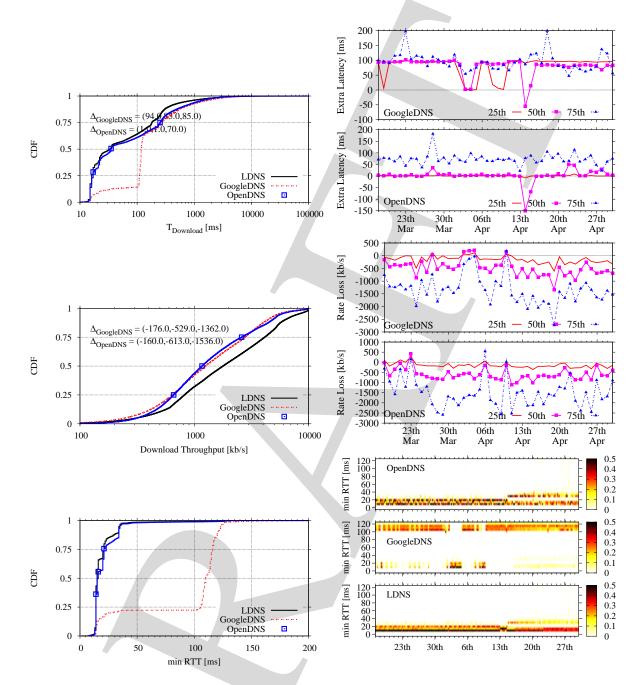


Figure 10: Edgecast metrics

ingly, differently from EdgeCast, changes in the DNS mapping are not reflected in shifts of the Δ indexes of $T_{Download}$. In fact, even if users are redirected to "close" data centers (e.g., RTT = 20 ms) the objects download time is still high (> 100 ms). To highlight this effect, Fig. 11 (bottom-left) reports CDF of $T_{Process}$, i.e., the time between the TCP ACK of the HTTP GET and the first packet sent by the server carrying the reply. This is an estimation of the time spent by the server processing the HTTP request. As we can see, this is typically higher than 100 ms, with Google DNS showing a slightly higher tendency to redirect to servers with higher responsiveness.

These results allow to formulate an hypothesis on the system architecture: "is it true that European Facebook servers

contacted are just front-end and requests are still sent to the U.S.?" In this case, the front-end are responsible for handling the TCP connection (that's why the RTT reflect the distance between client and server) but requests are still handled using backend in the U.S. (that's why download performance show "long" values even if content is served by close servers). We haven't found any strong conclusion about this aspect and further analysis are needed.

Considering Facebook static content metrics reported in Fig. 12, we can observe that LDNS resolver show better performance. In this case we can see the combination of two effects. On the one hand CDN replica server should be close to the users, so the download time should be related to the RTT. This is visible in the first part of the distribution

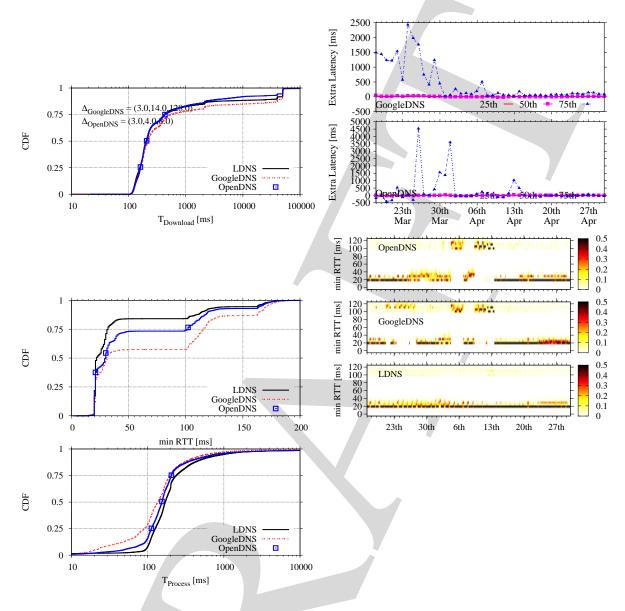


Figure 11: Facebook dynamic metrics.

which is related to the position of the Akamai caches. On the other hand, extra delays might be related to the frontend/back-end hypothesis previously introduced. Again, further analysis are needed to investigate on the subject.

Akamai-other: We conclude the analysis showing results related to other Akamai content, i.e., services and content which are not Facebook static. Overall, results are similar to the previous case. Considering $T_{Download}$, i.e., small objects, median values are quite similar and related to the minimum RTT. Instead, the tail of the distributions are quite different with LDNS present significant advantages. Considering larger objects, both public resolvers present worse performance than LDNS even if differences are more accentuated for OpenDNS.

6. FINAL REMARKS AND CONCLUSIONS

When assessing web browsing performance three compo-

nents need to be considered: (i) the DNS resolution latency, (ii) the quality of the mapping between hostnames and server IP addresses, and (iii) the performance offered by the network path and the contacted server. Among the three, our results indicate that the first component does not play an important role. In fact, the aggressive adoption of DNS prefetching (not captured in previously works) mask the gain of faster look up times.

When considering the interplay between CDNs and DNS resolvers, our results confirm previous studies showing that public DNS resolvers suffer of less optimal mapping policies. However, differently from previous studies, we quantified the performance differences over an extensive dataset. Results indicate that public resolvers lead to impairment for the download rate even higher than 50% with respect to LDNS. Performance-wise, we can conclude that there is no good motivation to opt for a public DNS resolver. This is true for the users of the considered ISP but we believe that similar

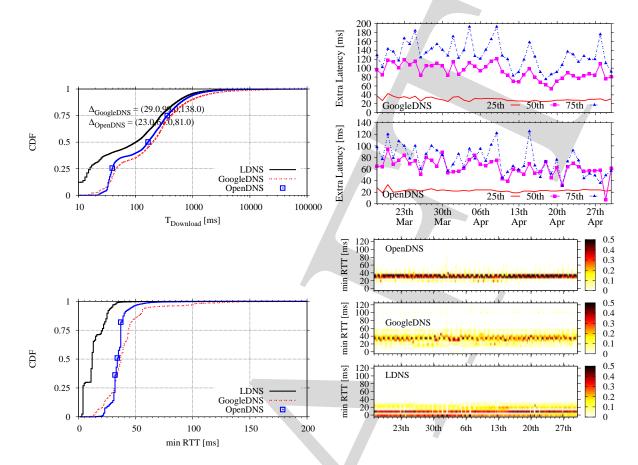


Figure 12: Facebook static metrics.

results should hold also for other ISPs assuming a reasonable provisioning of the local DNS resolvers.

Some users might still prefer to opt for public DNS resolvers because of specific features (e.g., parental control, malware protection, more reliable service, etc.). In such cases, they should be warned of possible performance impairment. Moreover, our study suggests that advanced mechanisms such as ECS are still not enough mature. This call for further optimization of the current DNS mapping mechanisms. Notice that this effort is not simply tackling the recursive resolvers design. It should instead address the more challenging improvement of the synergies between DNS recursive resolvers and CDNs.

Acknowledgments

The research leading to these results has received funding from the European Union under the FP7 Grant Agreement n. 318627 (Integrated Project "mPlane"). We would also like to thank Dr. Paul Mockapetris for the interesting discussion and suggestions.

APPENDIX

A. CDN RANKING

Table 3 reports the rank of the top 50 organizations/CDNs with respect to the fraction of flows, bytes and users. Results refer to the first week of data of the dataset.

The CDNs/organizations names are obtained using the MaxMind Organee database (see Sec. 3). Notice that some

processing is neede to group names together since each organization can be represented by multiple names.

2. REFERENCES

- J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, B. Weihl, "Globally Distributed Content Delivery", Internet Computing, IEEE, vol.6, no.5, pp.50,58, Sep/Oct 2002
- [2] V. Gehlen, A. Finamore, M. Mellia, M. M. Munafó, "Uncovering the Big Players of the Web", Traffic Monitoring and Analysis (TMA'12), Vienna, AT, April 2012
- [3] C. Huang, D. A. Maltz, J. Li, A. Greenberg, "Public DNS System and Global Traffic Management," IEEE INFOCOM, 2011.
- [4] S. Otto, M. A. Sánchez, J. P. Rula, F. E. Bustamante, "Content Delivery and the Natural Evolution of DNS: Remote DNS Trends, Performance Issues and Alternative Solutions", ACM IMC, 2012.
- [5] B. Ager, W. Mühlbauer, G. Smaragdakis, S. Uhlig, "Comparing DNS resolvers in the Wild", ACM IMC, 2010.
- [6] M. A. Sánchez, J. S. Otto, Z. S. Bischof, D. R. Choffnes, F. E. Bustamante, B. Krishnamurthy, W. Willinger, "Dasu: Pushing Experiments to the Internet's Edge", USENIX NSDI, April 2013.
- [7] J. S. Otto, M. A. Snchez, J. P. Rula, T. Stein, F. E. Bustamante, "Namehelp: Intelligent Client-side DNS Resolution", Poster at ACM SIGCOMM, 2012.
- [8] C. Huang, I. Batanov, J. Li, "A Practical Solution to the Client-LDNS Mismatch Problem", SIGCOMM Comput. Commun. Rev. 42, 2, pp. 35-41, March 2012.

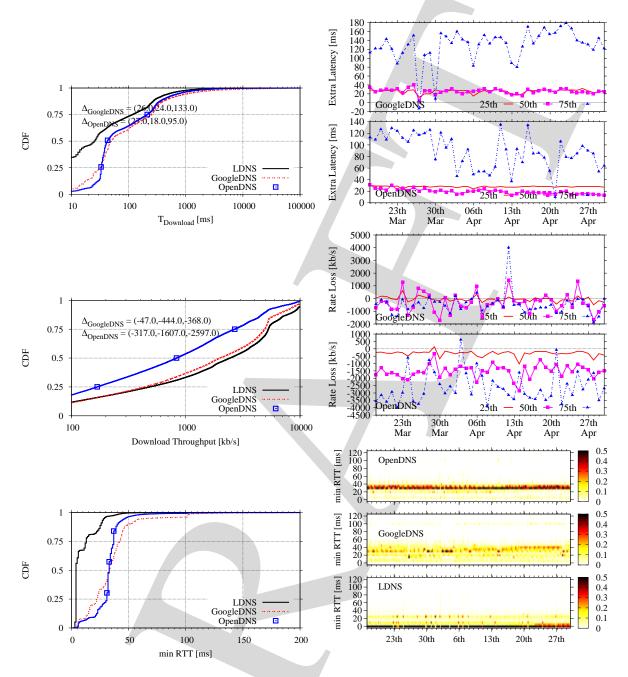


Figure 13: Akamai-other metrics.

- [9] C. Contavalli, W. van der Gaast, S. Leach, E. Lewis, "Client Subnet in DNS Requests", IETF Internet Draft draft-vandergaast-edns-client-subnet-02 (2013).
- [10] T. Callahan, M. Allman, M. Rabinovich, "On Modern DNS Behavior and Properties", ACM SIGCOMM Computer Communication Review, 43(3), July 2013.
- [11] R. Khosla, S. Fahmy, Y. C. Hu, "Content Retrieval using Cloud-based DNS", IEEE Global Internet Symposium, 2012.
- [12] P. Vixie, "What DNS is Not", Commun. ACM, 52(12):43-47, 2009.
- [13] G. Economou, "How Akamai Maps the Net: an Industry Perspective", 2011. http://www.akamai.com/dl/akamai/ economu_mapping_the_internet.pdf
- [14] F. Streibelt, J. Bottger, N. Chatzis, G. Smaragdakis,

- A. Feldmann, W. Willinger, "Unintended Consequences: Exploring EDNS-Client-Subnet Adopters in your Free Time" ACM IMC, 2013.
- [15] A. Finamore, M. Mellia, M. Meo, M.M.Munafò, D.Rossi, "Experiences of Internet Traffic Monitoring with Tstat," Network, IEEE, vol.25, no.3, pp.8,14, May-June 2011.
- [16] I. N. Bermudez, M. Mellia, M. M. Munafò, R. Keralapura, A. Nucci, "DNS to the Rescue: Discerning Content and Services in a Tangled Web", ACM IMC, 2012.
- [17] A. Finamore, I. Bermudez, M. Mellia, "Public DNS Resolvers: Friends or Foes? (Technical Report)", http://www.retitlc.polito.it/finamore/papers/ dns-techreport.pdf, August 2013.

Table 3: Top 50 CDN Rankings considering 1 week of data. **Rank by flows** | # | MaxMind Orgname | %F | %B | %U | | # | MaxMind Orgname | %F | %B | %U | | # | MaxMind Orgname | MF | %B | %U | | # | MaxMind Orgname | MF | %B | %U | | # | MaxMind Orgname | MF | %B | %U | | # | MaxMind Orgname | MF | %B | %U | | # | MaxMind Orgname | MF | MAXMIND | MAXMIND

#	MaxMind Orgname	%F	%В	%U
1	akamai	16.58	24.97	85.82
2	google	14.33	29.97	83.74
3	amazon	4.26	1.12	80.52
4	facebook	2.63	0.45	78.27
5	edgecast	2.02	1.74	
6	level.3	2.01	3.58	
7	yahoo	1.69	0.18	
8	ovh	1.62	1.74	
9	leaseweb	1.58	2.56	
10	microsoft	1.48	0.24	76.78
11	kataweb.s.p.a	1.37	0.23	33.20
12	aruba.s.p.a	1.31	0.88	66.98
13	limelight	1.24	5.52	56.29
14	avast.software.a.s.	1.04		
15	ISP	0.96	0.09	44.55
16	zynga	0.96		4.28
17	verizon	0.96	0.03	71.67
18	appnexus	0.78	0.02	62.11
19	italia.online.s.p.a	0.77	0.11	26.90
20	videotime.spa	0.76	1.39	20.33
21	itnet.s.p.agenova	0.73	0.14	46.49
	italia			
22	on.line.services	0.70	0.04	45.16
	.for.advertising, .mar-			
	keting,.media			
23	hetzner.online.ag	0.69	0.23	59.58
24	criteo.sa	0.67	0.02	64.07
25	pb_web_media_b.v.	0.66	0.05	7.02
26	adnexus	0.64	0.02	61.01
27	shiny.corporation	0.60	0.01	52.89
28	telecitygroup. interna-	0.54	0.03	67.00
	tional.limited			
29	il.sole.24.ore	0.48	0.07	18.07
30	24/7.real.media	0.44	0.02	63.85
31	level.ip.italia.s.r.l.	0.44	0.03	56.16
32	deutsche.telekom.ag	0.43		49.54
33	isprime	0.43	0.03	21.73
34	wikimedia.foundation	0.42	0.06	38.57
35	bt.italia.s.p.a.	0.41	0.11	
36	ISP1	0.38	1.29	31.20
37	reflected.networks	0.38		26.80
38	cdnetworks	0.36	0.37	57.66
39	quantcast.corporation	0.33	0.00	52.72
40	openx.technologies	0.33	0.00	44.87
41	infracom.italia.s.p.a.	0.31	0.06	24.53
42	haldex.ltd	0.31	0.91	
43	kpnqwest.italia.s.p.a.	0.29	0.06	
44	addthis	0.29	0.00	
45	vkontakte.ltd	0.27	2.37	8.28
46	cloudflare	0.26	0.17	
47	tinet.spa	0.26	0.53	58.16
48	verisign.global. reg-	0.25	0.01	71.50
49	istry.services badoo.limited	0.25	0.05	3.79
50		0.25	0.05	
30	saferoute.incorporated	0.25	0.01	33.92

#	MaxMind Orgname	%F	%B	%U
1	google	14.33	29.97	83.74
2	akamai	16.58	24.97	85.82
3	limelight	1.24	5.52	56.29
4	level.3	2.01	3.58	68.77
5	leaseweb	1.58	2.56	60.66
6	vkontakte.ltd	0.27	2.37	8.28
7	ovh	1.62	1.74	65.72
8	edgecast	2.02	1.74	75.31
9	videotime.spa	0.76	1.39	20.33
10	ISP1	0.38	1.29	31.20
11	amazon	4.26	1.12	80.52
12	m247.ltd	0.01	1.08	4.63
13 14	haldex.ltd	0.31 1.31	0.91	15.25 66.98
15	aruba.s.p.a	0.43	0.88	49.54
16	deutsche.telekom.ag tripartz.b.v	0.43	0.86	17.05
17	eweka.internet. ser-	0.00	0.72	6.35
1 '	vices.b.v.	0.00	0.12	0.00
18	webazilla.b.v.	0.25	0.64	16.07
19	tinet.spa	0.26	0.53	58.16
20	ddl.technologies.srl	0.02	0.52	0.38
21	cogent.communications	0.17	0.51	26.61
22	fdcservers.net	0.07	0.51	25.42
23	facebook	2.63	0.45	78.27
24	dailymotion.s.a.	0.12	0.43	9.13
25	kpt.network.b.v.	0.00	0.39	0.55
26	cdnetworks	0.36	0.37	57.66
27	dstorage.s.a.s.	0.01	0.32	0.21
28	highwinds.network.	0.08	0.30	29.56
29	group hosting.services	0.21	0.29	56.92
30	advancedhosters. lim-	0.21	0.29	11.54
30	ited	0.10	0.25	11.54
31	at&t	0.02	0.27	9.20
32	ip.transit	0.01	0.26	4.48
33	microsoft	1.48	0.24	76.78
34	private.layer.inc	0.05	0.24	5.63
35	hetzner.online.ag	0.69	0.23	59.58
36	kataweb.s.p.a	1.37	0.23	33.20
37	atrato.ip.networks	0.04	0.23	19.37
38	link11.gmbh	0.00	0.20	0.93
39	yahoo	1.69	0.18	68.24
40	cloudflare	0.26	0.17	56.31
41	kaia.global.networks.lto		0.17	0.64
42	lemuria.communication	8 0.01	0.16	0.97
44	advancedhosters.limited nforce.entertainment.b.		0.16	9.79 10.40
45	voxility.srl	0.01	0.15	4.11
46	garr-	0.01	0.15	2.34
13	b.backbone.and.pops	0.00	0.10	2.04
47	voxility.s.r.l.	0.03	0.15	5.39
48	jsc.ukrtelecom	0.00	0.14	0.59
49	itnet.s.p.agenova	0.73	0.14	46.49
1	italia			
50	italian.public. broad-	0.05	0.13	10.91
	casting. company			

	Rank by users					
#	MaxMind Orgname	%F	%B	%U		
1	akamai	16.58	24.97	85.82		
2	google	14.33	29.97	83.74		
3	amazon	4.26	1.12	80.52		
4	facebook	2.63	0.45	78.27		
5	microsoft	1.48	0.24	76.78		
6	edgecast	2.02	1.74	75.31		
7	verizon	0.96	0.03	71.67		
8	verisign.global. reg-	0.25	0.01	71.50		
	istry.services					
9	level.3	2.01	3.58	68.77		
10	yahoo	1.69	0.18	68.24		
11	telecitygroup. interna-	0.54	0.03	67.00		
	tional.limited					
12	aruba.s.p.a	1.31	0.88	66.98		
13	ovh	1.62	1.74	65.72		
14	criteo.sa	0.67	0.02			
15	24/7.real.media	0.44	0.02	63.85		
16	addthis	0.29	0.00			
17 18	appnexus adnexus	0.78	0.02			
19	leaseweb	1.58	2.56	60.66		
20	hetzner.online.ag	0.69	0.23			
21	internap.network. ser-	0.09	0.23	58.82		
21	vices.corporation	0.20	0.02	36.62		
22	omniture	0.24	0.00	58.53		
23	top-half-as	0.22	0.00	58.49		
24	tinet.spa	0.26	0.53			
25	cdnetworks	0.36	0.37	57.66		
26	siminn.danmark.a/s	0.24	0.02			
27	hosting.services	0.21	0.29	56.92		
28	fastly	0.20	0.03			
29	datapipe	0.19	0.01	56.45		
30	cloudflare	0.26	0.17	56.31		
31	limelight	1.24	5.52	56.29		
32	level.ip.italia.s.r.l.	0.44	0.03			
33	weborama.sa	0.17	0.00	53.78		
34	shiny.corporation	0.60	0.01	52.89		
35	quantcast.corporation	0.33	0.00			
36	the.rubicon.project	0.14	0.01	51.86		
37	ip.exchange.gmbh	0.20	0.08			
38	deutsche.telekom.ag	0.43	0.88	49.54		
39	fullsix.spa	0.23	0.01	49.29		
40	aufeminin.aufeminin.co		0.01	48.85		
41	bt.italia.s.p.a.	0.41	0.11	48.66		
42	ccanet.limited	0.08	0.01	47.70		
43	itnet.s.p.agenova .italia	0.73	0.14	46.49		
44	jay.net.a/s	0.06	0.00	45.94		
45	godaddy.com,.llc	0.00	0.06			
46	on.line.services.for.	0.70	0.04	45.16		
10	advertising, .market-	0.70	0.04	10.10		
	ing,.media					
47	tiscali.b2b	0.19	0.02	45.02		
48	openx.technologies	0.33	0.00	44.87		
49	ISP	0.96	0.09			
50	colt.telecom	0.25	0.00			