June 2026
Geoff Huston

DNS Query Duplication

The tendency for free resources to be exploited to the point of over-consumption and even exhaustion is a well-known economic and social concept, frequently described with the term *the tragedy of the commons*. Because use of the resource costs each individual user nothing, there is no financial incentive for any user to moderate their consumption of the common resource. Over-consumption inevitably ensues!

In the DNS, name resolution space queries are free. To what extent do we see over-querying on the part of recursive resolvers in the DNS? There is certainly some motivation for a recursive resolver to query enthusiastically, given that a strategy of what one could term *over-querying* can work around slow or unresponsive nameservers, or repair UDP datagram loss. O*ver-querying* describes a behaviour where a DNS resolution system does not wait for a local timer to expire to conclude that an original UDP-based query has been lost but repeats the query well before a reasonable timeout interval has elapsed. The apparent thinking behind this *rapid-fire over-querying* behaviour is that in the event of query loss, the subsequent query may be successful, and the response to this subsequent query will arrive back to the resolver soon after the response to the original query would've arrived, saving the resolver (and the end user) waiting for a query timeout interval. If both queries are successful, then the resolver will simply discard the responses to the later query. As queries are free, the cost to the authoritative server in generating the answer twice is borne by the authoritative server and not the recursive resolver. Such a behaviour can improve the responsiveness and resilience of a recursive resolver, and the cost of generating these additional responses is met by others. In other words, if there is no cost to the resolver in this behaviour then there is no motivation to adopt more conservative query stance!

The question here is: To what extent are DNS resolvers in today's Internet overly profligate in generating duplicate queries?

We'll use the APNIC Labs measurement data generated on a single day, the 1st of June 2026, for this study. This system uses an online ad presentation system to seed measurement tasks on a collection of users' browsers. The system generates some 35 million ad presentations for each 24-hour period, spread over users drawn from all parts of the Internet. Each individual measurement ad presentation contains up to 15 different component URL fetches (for different measurement tasks), where each URL contains a unique DNS name to resolve. The use of a unique label ensures that no DNS caching is used to resolve these names, so all end user DNS resolution requests are seen as queries at the measurements DNS authoritative servers and not answered by intermediate DNS caches held by resolvers.

There are various forms of DNS behaviours that are configured into the APNIC measurement set, some of which are deliberately intended to generate additional queries (such as, for example, a DNS name where the authoritative nameserver always returns a SERVFAIL response code). Here, we are looking at just those DNS query names which are served by a dual stack nameserver that always promptly responds to A and AAAA query types. UDP is not a reliable transport protocol, and some datagram loss is expected, but it would be reasonable to anticipate an overall query loss rate of some 2% or lower in most circumstsances, so that we expect to see a comparable query duplication rate when looking at a random sample set of DNS data from this experiment.

The measurement experiment directs users located in each of these regions to a DNS server that is physically located in the same region as the user. We use a division of the Internet into six major areas, as shown Table 1.

Code	Region	Server Location
ap	East Asia/Oceania	Singapore
am	North America	Dallas
la	Latin America	Sao Paulo
eu	Europe and Western Asia	Frankfurt
in	Indian Subcontinent	Mumbai
hk	China, Hong Kong and Macao	Hong Kong SAR

Table 1 – APNIC Lab's Measurement Regions

The data set that was gathered during this 24-hour period is shown in Table 2. Here, we are distinguishing between an original query, and successive duplicate queries, where a *duplicate* is defined as a query that has the same query name and query type.

Region	Exps	Unique qnames	Non-Dup qnames	Dup qnames	Dup qname Rate	Dup Queries	Total Queries	Dup Query Rate
ap	10,615,195	48,109,644	30,106,040	18,003,604	37%	37,843,955	85,953,599	44%
eu	8,189,624	36,891,297	26,747,916	10,145,381	27%	17,979,708	54,871,005	32%
in	5,774,302	26,242,263	10,348,340	15,893,923	61%	36,112,196	62,354,459	57%
am	4,016,880	20,707,638	14,965,571	5,742,067	28%	9,416,280	30,123,918	31%
la	2,305,659	10,212,137	6,732,592	3,479,545	34%	7,002,049	17,214,186	40%
hk	1,198,329	4,330,936	2,280,032	2,050,904	47%	4,763,721	9,094,657	52%
Total	32,099,989	146,493,915	91,180,491	55,313,424	38%	113,117,909	259,611,824	43%

Table 2 - Query Duplication Measurement Results

While the average for the full set of data for this day shows a query duplication rate of 38% of all seen query name/query type couplets, users in the **in** (Indian Subcontinent) region have a far higher query duplication rate of 61%, and the **hk** (China/HK) region shows a 47% query duplication rate. The **am** (North America) has a query name duplication rate of 28%, and **eu** (Europe) has a 27% query name duplication rate. This data indicates that there is a more prevalent DNS resolution behaviour that duplicates DNS queries in the Indian and Chinese regions.

Of course, average values can be misleading, and average values can be skewed by a small number of DNS resolution environments performing a large number of duplicate queries. The highest number of duplicate queries was 8,771 duplicate queries in this 24-hour observation period, observed at the EU server in Frankfurt. The overall distribution of the number of duplicates of each of the queries that are part of this measurement are shown in Figure 1.

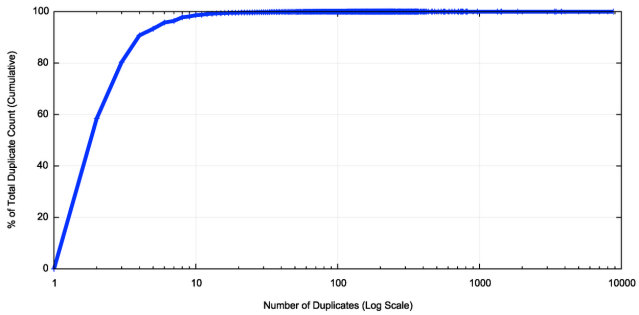

Figure 1 – Distribution of query duplicate counts

Some 90% of all duplicate query sets for a single qyery name / query type couplet occurs 3 or fewer times, and 58% of duplicates are observed as a single duplicate of the original query. Just 1% of duplicate

query sets are sets with 10 or more duplicate queries. The most common form of duplicate query behaviour is where the original DNS query is duplicated just once.

A DNS resolver using a UDP-based query will conventionally perform a repeat query if a UDP timer expires, as it is attempting to recover from the situation where the sender is assuming that the UDP query or response datagram has been dropped. A DNS recursive resolver will also perform a re-query when the earlier DNS response has been cached and the response's cache TTL (Time To Live) time has expired. We can discern between these two resolver behaviours by looking at the elapsed time between the original query and the duplicate query.

Figure 2 shows the duplicate query profile over the 24-hour period looking at the elapsed time between the original query and the duplicate query. The time base used here is a unit of minutes.

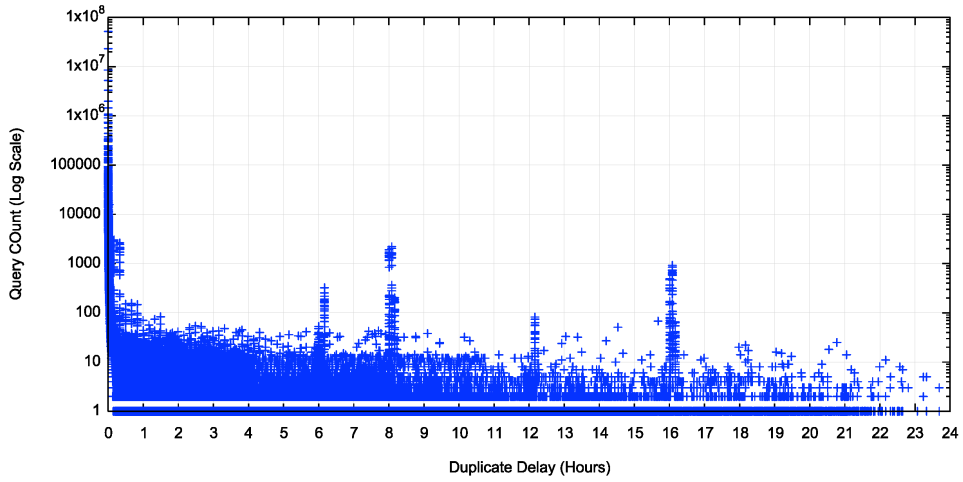

Figure 2 – Time distribution of duplicate queries (hours)

As well as an extremely large collection of duplicates in the period immediately following the initial query, there are visible local peaks in duplicate queries at periods of 6, 12 and 18 hours after the initial query, and a second repeat pattern at 8 and 16 hours. The TTL value of DNS responses in this measurement is set to 60 seconds, so these far longer periods associated with duplicate queries is likely to be some form of resolver implementation behaviour where it is attempting to perform a refresh of earlier cached data that was retrieved some hours earlier.

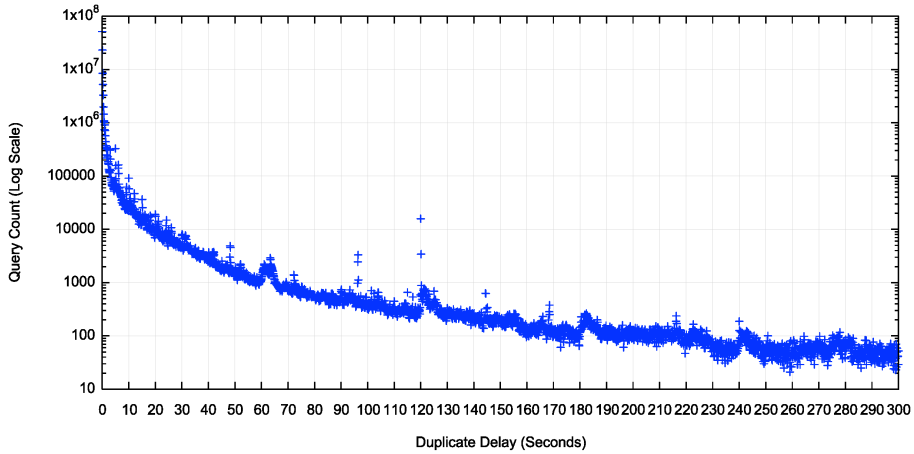

Figure 3 – Time distribution of duplicate queries (seconds)

Figure 3 looks at the first 300 seconds between the initial query and the duplicate query, using a unit of queries per second. In this figure there are local peaks at 60, 120, 180 and 240 seconds, which is likely to be related to TTL expiration in the resolver's cache. The cache refresh is supposed to be triggered by an incoming query for the cached query name and query type. However, the nature of this measure is

that each DNS query name is uniquely generated for a single use by a single user. There are not supposed to be duplicate queries, and there are not supposed to be query-triggered cache refresh queries. It is still the case here that the number of these longer-period duplicates is far smaller than the number of duplicate queries that occur in the first second.

Figure 4 looks at the query duplication profile in the first second following the initial query, using a millisecond (ms) granularity.

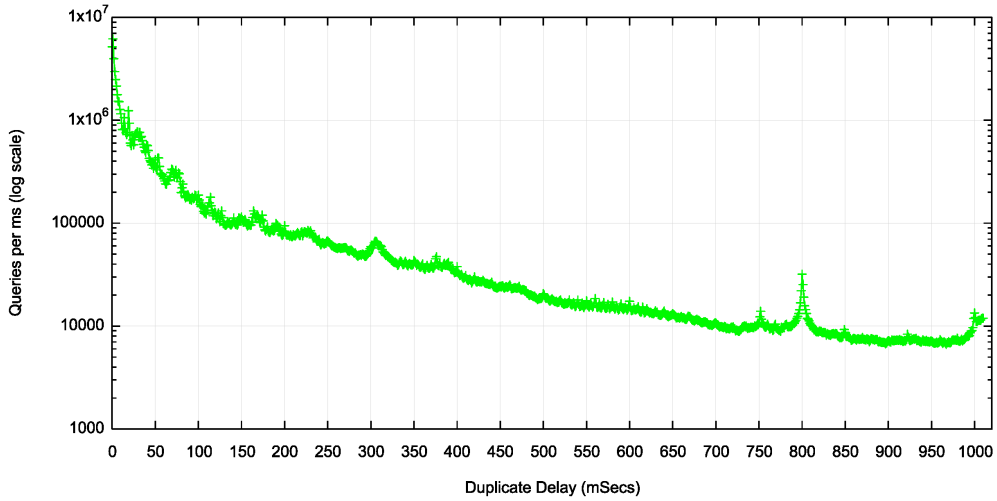

Figure 4 – Time distribution of duplicate queries (milliseconds)

Here there are local peaks at 320, 750 and 800ms, as well as a 1 second peak.

Some 45% of all duplicate queries occur in the first 10ms following the initial query (Figure 5). This indicates that the major component of the duplicate DNS query behaviour we are observing here is not due to re-querying following UDP-related timeouts in the part of the sender. This is the result of some other resolver behaviour mode that is observed at the authoritative server as a rapid-fire couplet of queries for the same query name and query type (Figure 6).

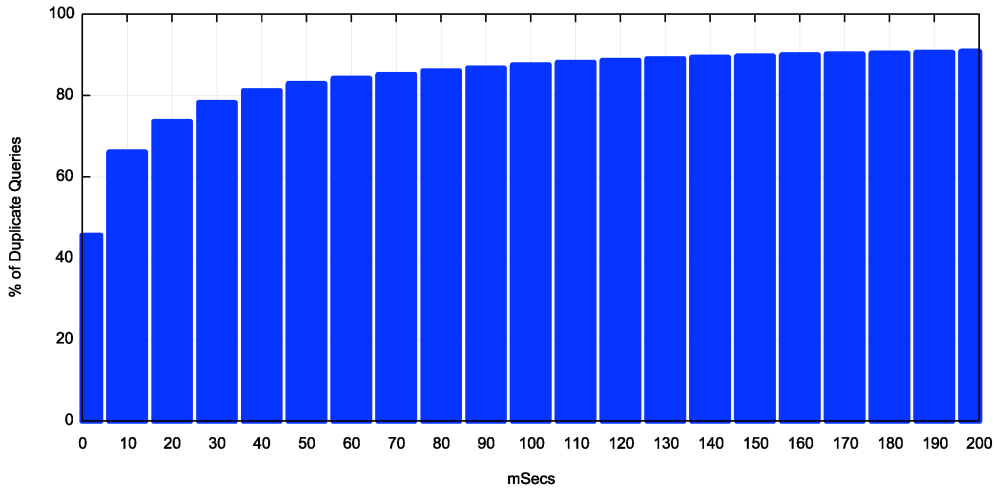

Figure 5 – Cumulative distribution of duplicate queries (ms)

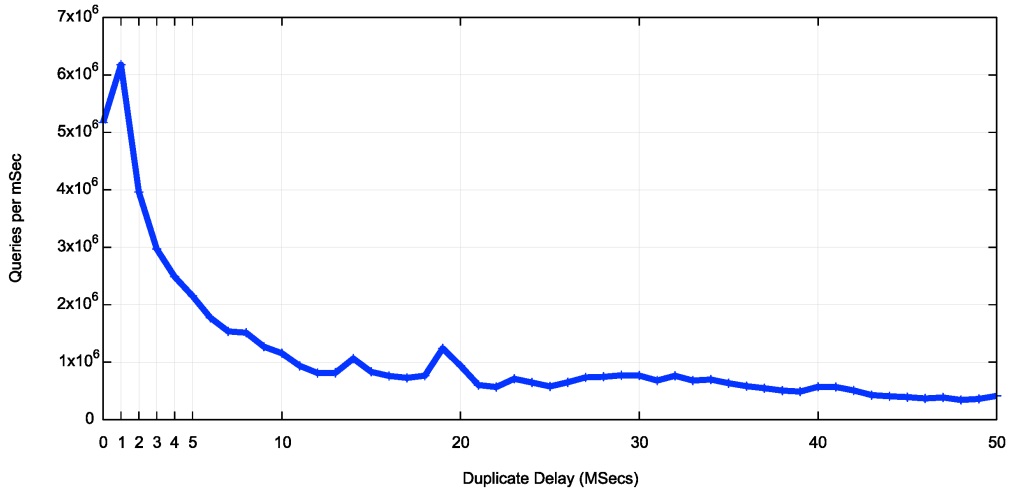

Figure 6 – Time distribution of duplicate queries (ms)

An interpretation of this behaviour is that the resolver could be attempting to improve the performance and resilience of the DNS query process by sending the two queries back-to-back using some element of difference that improves the overall resilience of the process. For example, a resolver could generate queries to two different authoritative nameservers at the same time, and it could then act upon the first nameserver to respond. Or it could use IPv4 and IPv6 transports. Or the resolver could both forward the query to an open recursive resolver and attempt to resolve the name itself.

However, we only have one authoritative nameserver in this experiment, so using multiple authoritative nameservers in some form of resolution race is not an option here. We can measure the diversity of resolver IP addresses generating the queries in the first 100ms after the initial query. Is the query being passed to multiple recursive resolvers to set up a race between resolvers to resolve a name in an effort to optimise resolution performance and resolution resilience? Or is there large volumes of a degenerate form of query duplication where the same recursive resolver simply duplicating the query and sending it to the same authoritative nameserver in a back-to-back manner?

For this one day where we observed 146,493,915 unique query names, and looking only at the window of the first 100ms after the initial unique query, we saw 68,589,808 duplicate queries. The distribution of the number of duplicates of each query is shown in Figure 7. While the maximum number of duplicate queries in this 100ms interval is 120 queries, in 90% of cases the duplicate query count is 1 or 2.

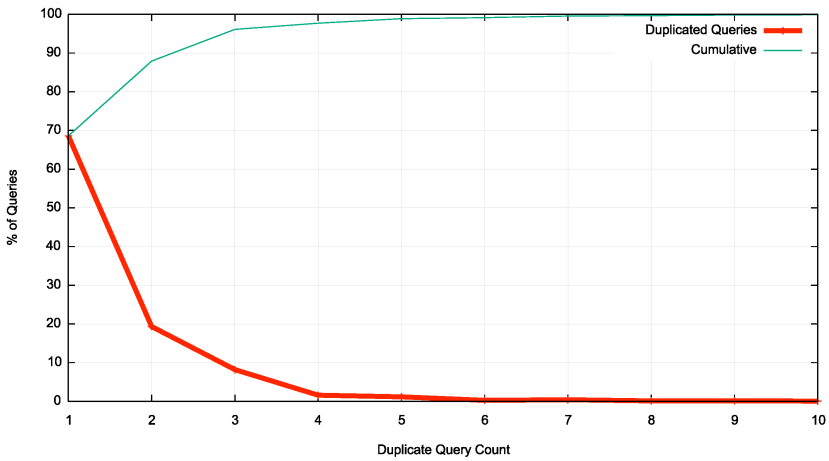

Figure 7 –Distribution of duplicate query counts within 100ms

Of these queries we observed 56,498,331 using a different IP address than the initial query, which is 82% of these duplicate queries, and 12,091,477 using the same IP address as the initial query. We observed that 48% of the duplicating query sets are IPv4 only, 12% are IPv6 only and the remaining 40% have

both IPv4 and IPv6 queries. It's possible to understand the motivation of a strategy of performing a rapid query duplication by asking on both IPv4 and IPv6. It's not exactly friendly to the network or to the authoritative server, as it gratuitously doubles the authoritative server's query load, but from a strictly selfish perspective it can add a level of resilience and fallback performance to the resolution operation.

In terms of the networks that host these recursive resolvers, 3,572,156 query sets (5% of the total) have directed their queries to multiple recursive resolvers that are located in multiple networks (different origin ASes), while 41% use resolvers in just one network, using the same network as the end client. Some 11,167,864 query sets (16% of the total) use a known open resolver. Again, this rapid splaying of queries to multiple recursive resolvers can be self-justified as a measure intended to improve the resilience and performance of a local DNS resolution system, while it exacts a cost in increased query loads on the authoritative servers, as it doubles the query load.

The DNS resolution behaviour that is challenging to justify is the rapid query duplication being performed by the same resolver IP address to ther same authoritative server. This behaviour appears to be quite commonplace, with 6,117,444 query sets from a total of 44,810,600 query sets (14%) that were observed to have this form of single resolver query duplication.

Which networks host these duplicating resolvers? Table 3 lists the 25 networks that host resolvers with the highest total of same-resolver rapid query duplication levels within each network.

AS	% Dups	Dup Qnames	Total Qnames	AS Name
55836	14%	3,090,635	20,654,445	Reliance Jio Infocom, IN
9498	27%	1,236,896	4,435,556	BHARTI Airtel, IN
7552	13%	773,867	5,545,325	Viettel Group, VN
45899	27%	761,806	2,760,804	VNPT, VN
23693	49%	682,152	1,377,963	Telekomunikasi Selular, ID
13335	3%	369,394	10,157,682	Cloudflare, US
3320	60%	289,198	479,660	Deutsche Telekom AG, DE
17676	11%	181,537	1,527,095	GIGAINFRA - SoftBank Corp., JP
4804	58%	167,011	285,487	Optus, AU
17639	18%	154,817	856,481	Converge ICT Solutions, PH
327931	90%	152,990	168,395	Optimum Telecom Algeria, DZ
8881	29%	148,564	507,695	Versatel GmbH, DE
7303	51%	147,629	285,956	Telecom Argentina, AR
4818	13%	136,752	1,006,486	DiGi Telecommunications, MY
36924	92%	113,588	123,058	GVA Cote d'Ivoire, CI
15169	0%	111,535	13,214,739	Google, US
199739	25%	107,810	415,456	Earthlink-DMCC-IQ, AE
16509	39%	106,335	267,895	Amazon-02, US
29465	68%	97,586	141,644	VCG-AS - MTN, NG
6697	18%	91,813	509,262	Beltelecom, BY
855	98%	85,942	87,399	Bell Canada, CA
9269	59%	85,799	144,114	Hong Kong Broadband, HK
4685	20%	78,896	393,360	Asahi Net, JP
42772	64%	78,564	121,660	Unitary enterprise A1, BY
17882	27%	78,433	290,080	Univision, MN

Table 3 - Query Self-Duplication Rates – Top 25 Networks (Full Table: https://www.potaroo.net/ispcol/2026-06/table3.data)

The open resolvers, operated by Google and Cloudflare, are listed here, but with overall duplication rates of 0% and 3%, this level of query duplication is relatively minor. Of greater concern are resolvers located in Reliance Jio and Bharti Airtel in India, Viettel and VNPT in Vietnam, and Telekomunikasi Selular in Indonesia. The high query duplication rates seen in the DNS resolution systems used by users served by Deutsche Telekom in Germany, Softbank in Japan, Amazon O2, and Bell Canada are also curious. These are all very high-volume DNS resolution environments.

The top 25 individual resolvers that are generating the largest volume of these rapid-fire self-replicated queries are shown in Table 4.

Resolver IP	% Dups	Dup Qnames	Total Qnames	AS	AS Name
113.164.250.130	39%	139,345	356,264	45899	VNPT, VN
113.164.250.138	38%	137,652	355,298	45899	VNPT, VN
114.125.0.95	87%	125,056	142,499	23693	Telekomunikasi Selular, ID
114.125.0.94	87%	123,958	141,345	23693	Telekomunikasi Selular, ID
114.125.64.102	82%	85,126	102,640	23693	Telekomunikasi Selular, ID
171.241.119.236	21%	84,464	389,492	7552	Viettel, VN
171.241.119.228	21%	83,912	388,751	7552	Viettel, VN
114.125.160.102	83%	76,331	91,668	23693	Telekomunikasi Selular, ID
114.125.160.103	85%	73,118	85,864	23693	Telekomunikasi Selular, ID
114.125.64.103	83%	70,395	84,368	23693	Telekomunikasi Selular, ID
103.162.137.37	98%	65,859	66,617	142127	Fibercom Technologies, PK
182.0.193.30	75%	65,075	86,451	23693	Telekomunikasi Selular, ID
213.222.178.78	99%	62,423	62,720	21334	One Hungary Ltd., HU
182.0.193.31	76%	61,922	81,172	23693	Telekomunikasi Selular, ID
113.164.250.202	23%	59,316	256,859	45899	VNPT, VN
113.164.250.210	23%	58,829	255,412	45899	VNPT, VN
171.225.222.226	31%	58,690	188,971	7552	Viettel, VN
171.225.222.234	30%	58,093	188,100	7552	Viettel, VN
171.225.222.235	30%	57,007	184,137	7552	Viettel, VN
171.225.222.227	30%	56,467	183,099	7552	Viettel, VN
41.200.0.250	92%	55,073	59,443	327931	Optimum Telecom Algeria, DZ
171.225.222.228	38%	53,263	137,638	7552	Viettel, VN
171.225.222.236	38%	53,211	137,722	7552	Viettel, VN
185.239.18.26	33%	52,894	157,192	199739	Earthlink-DMCC-IQ, AE

Table 4 - Query Self-Duplication Rates – Top 25 Resolver IP addresses (Full Table: https://www.potaroo.net/ispcol/2026-06/table4.data)

Most of the resolvers in this table do not perform this rapid query duplication in every case, although the resolvers used in Fibercom Pakistan, One Hungary, and Optimum Telecom Algeria exhibit this behaviour in more than 90% of observed cases.

It's not clear what is causing this DNS query behaviour. It's unlikely that this is the result of a deliberate configuration choice, so a more likely explanation is that this is anomalous behaviour associated with the front end of a DNS *resolver farm*. Such DNS *resolver farms* are commonly used by access ISPs, as they allow a large-scale DNS resolution system to be implemented across multiple DNS engines, while simplifying the end clients' configuration by pushing out a small set of DNS resolver IP addresses that reference the load-balancing front-end machinery of the farm. One explanation of this observed duplication behaviour is that incoming DNS queries, which are normally passed to just one instance of a local DNS resolver engine in the farm, are being passed to multiple resolver engines, possibly by accident, or as a result of some transient overload condition or similar unintentional behaviour. It is unlikely that this failure will be visible to either the end user or the network operator, as the failure is not a case of inability to resolve a DNS name, or to provide an incorrect response, but a case where more queries than are normally required are being generated. If this is an intermittent behaviour for some DNS resolution systems, as appears to be the case in the collected data, then detection of this behaviour is highly unlikely.

It's often said of the DNS that it's a system that simply could not work in any usable manner were it not for caching. Not only does caching suppress DNS query volumes and improves DNS response times, but caching can also mask over many forms of otherwise anomalous resolver behaviour. We are amplifying this behaviour in our measurement experiment by using unique DNS labels, bypassing local DNS caches. This rapid duplication of a query that arrives at an authoritative nameserver would appear only at intervals bounded by the TTLs using the DNS data, and it is unlikely this this would form a significant portion of normal DNS traffic volumes coming from users, and as such it's a DNS resolution behaviour that is somewhat unlikely to be detected.

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston AM, M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net