The ISP Column
A column on things Internet

July 2024
Geoff Huston

Privacy and DNS Client Subnet

There has been a fundamental change in the architecture of service and content delivery over the Internet over the past decade. Instead of using the network to bring the remote user to a server that delivers the content or service, the content (or service) is loaded into one or more Content Delivery Network (CDN) platforms, which brings replicated instances of the content closer to each user. In networking terms if distance equates to cost, shaving the cost of millions of user content interactions by reducing the distance between the user and the content more than offsets the cost of replicating the content to hundreds or thousands of points in the CDN platform.

To ensure service consistency these replicated instances of the content are named with the same DNS name, and the DNS conventionally offers the same resolution outcome to each user when they query for the IP address of the content server. How can the CDN "steer" each user to the closest instance of the desired content to optimise the subsequent content transaction? At the same time the user is revealing their location within the network to inform this steering decision. To what extent is such a steering function compromising the privacy expectations of users with respect to the location and their online actions?

Routing Anycast

One approach to content steering is to use *routing anycast,* where every instance of the service uses a common IP address. It is left to the routing system to pass client's packets to the closest instance within the anycast service domain. The advantage of this approach lies in its simplicity for the CDN operator. As each instance of the CDN platform is announced to the Internet's Inter-Domain routing system using the Border Gateway Protocol (BGP), this advertisement will appear to be the most preferred route within the metrics of BGP Path selection in those parts of the network that are close to the point of announcement. Existing routes will remain locally preferred for more remote parts of the network.

> Anycast has been used extensively in the provision of the Root Zone DNS service, and as of June 2024 the root server system consists of 24 distinct anycast service addresses, and some 1,844 individual service instances (https://root-servers.org/).

The user is not revealing any additional information to the network infrastructure through this approach. The routing system passes the user's traffic to the closest instance of the replicated service and the user's location is not being passed into the infrastructure over and above the normal operation of the network's routing system.

The resolution of this anycast approach in mapping users to the "closest" instance of the service cloud depends on the resolution properties of the BGP protocol, the way that each local network operator configures their routing preferences, and the topology of the inter-domain Internet. If the network were constructed of a large collection of densely interconnected local networks, then the BGP route selection process would produce outcomes that would be well aligned to conventional concepts of proximity (Figure 1).

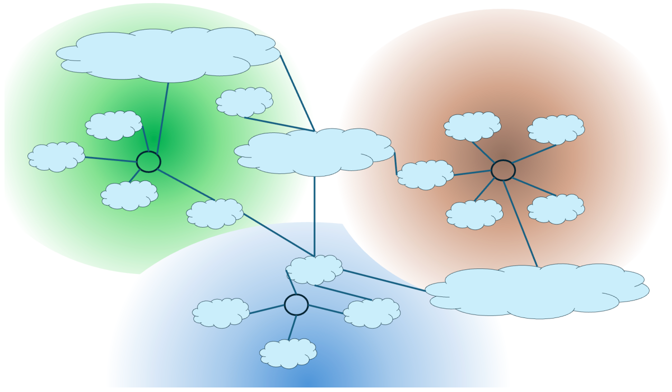

Figure 1 – Service Selection via Routing Anycast

The Internet is constructed of a small set of large transit providers in its "core" and a far larger set of attached satellite networks. The long-term average AS Path Length in the inter-domain routing space is around 4 ASes, and BGP route selection is relatively coarse as a result. As a recent measurement paper concludes: "… mobile clients are very frequently mapped to a geographically suboptimal anycast replica, with a significant impact on performance in terms of latency. We found that the long distances between clients and their assigned anycast server are not due to a lack of better, closer unicast replicas, and that this phenomenon is not bound to specific regions or particular ASes."[1]

Routing anycast is simple to operate for the service provider but performs at its best when the number of replicated service instance is both high (relative to the number of client networks) and widely dispersed.

> This is the line of reasoning used by Cloudflare with the 1.1.1.1 open DNS resolver. They have made the case that the density of deployment of the open resolver service is sufficiently high that the routing anycast used to reach a 1.1.1.1 service will reach a server that is close to the client, so the location "distortion" introduced by using 1.1.1.1 is minor as compared to the potential privacy leakage introduced by adding ECS to DNS queries.

There are also issues related to large span client networks, where the network's routing policies may prefer to use a single egress point, creating inefficient extended transit paths for remote users.

Can a CDN improve on this outcome?

Application Server Steering

One potential approach is by making the connection to the CDN a two-step process. In the first instance the user's application would be connected to any instance within the CDN, and the initial step is to compare the assumed location of the user to the CDN's set of service locations and then redirect the user's application to continue the transaction on a closer CDN service location using location specific service names. This approach is effective when the content transaction is sufficiently large, so that the added overheads in this initial service redirection can be amortised over the subsequent content transaction.

[1] S. Wassermann, J. P. Rula, F. E. Bustamante and P. Casas, "Anycast on the Move: A Look at Mobile Anycast Performance," *2018 Network Traffic Measurement and Analysis Conference (TMA)*, Vienna, Austria, 2018, pp. 1-8, doi: 10.23919/TMA.2018.8506570.

In this approach the user's location is calculated by the application, and not by the network's infrastructure, so the level of privacy compromise relating to the location of the user is limited to the application and is not passed over to the network infrastructure.

> This approach is used by Netflix, performing service routing within the application. The Netflix service addresses geolocate to a data server located in Oregon, in the US. This makes no sense for a client that is located anywhere other than Oregon. But the Netflix service is not a monolithic service. Each element of a video stream is effectively broken up into chunks and the application in the client side interacts with the server side to determine the optimal location for serving each chunk, using a server that is "close" to the user. In Netflix's terminology this approach is termed "Open Connect".

DNS Steering

However, this application server-based approach is not necessarily efficient for short transactions. A more efficient approach appears to lie in using the DNS itself to perform the selection of the optimal server instance. When the authoritative DNS server for the named service receives a query, it will provide an IP address that points to the server instance that appears (to the authoritative DNS server) to be the closest to the assumed location of the end client.

Of course, the IP address of the end client does not normally appear in DNS query passed from a recursive resolver to an authoritative server. The server needs to make a critical assumption at the outset, namely that the recursive resolver used by the end client is sufficiently close to the end client that optimising the response to the assumed location of the recursive resolver is the same as the occluded location of the end client.

This assumption, that a user's recursive resolver is located close to the user, is not always a good assumption. When the user is sending their queries to an open DNS recursive resolver, then some further considerations come into play.

If the open resolver service uses multiple points of presence, then the steering of the user to the closest instance of the open DNS recursive resolver is through anycast. The denser the anycast deployment, then the closer the DNS resolver will be to the end user, and the service address used by the recursive resolver to query the authoritative server will provide a reasonably good indication of the location of the end user.

When the open resolver deployment is sparse, then it is more likely that the user and the anycast-selected DNS resolver will be distant from each other, and the more likely that DNS-based steering will generate poor outcomes (Figure 2).

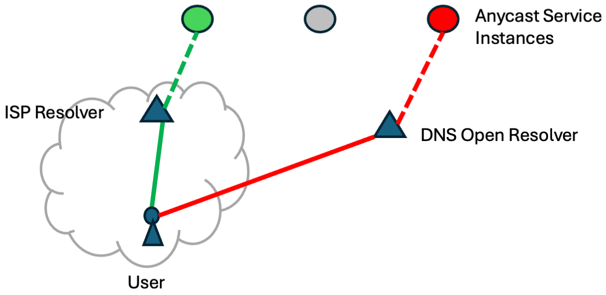

Figure 2 – Service Selection via DNS Steering

One potential response in such scenarios is for the recursive resolver to add an informational field to the DNS queries that it makes to authoritative servers for the anycast service that provides a useful clue as to the whereabouts of the end user. That way the DNS result can use the location of the end user to determine the appropriate DNS response, rather than assume that the DNS resolver and the end users are located sufficiently "close" to each other.

EDNS Client Subnet

In 2016 the IETF published RFC 7871, "Client Subnet in DNS Queries". This EDNS (RFC 6891) option allows the recursive resolver to attach the encompassing subnet of the client address received in the client-side query to the query that the recursive resolver passes to the authoritative server. There is provision for the authoritative server to use this client subnet in its response to indicate the network scope for which this response is intended. The recursive resolver attaches the response subnet to its cached responses and can then use its local cache where there is a match for both the query name and the query source IP address against the cached entry.

This is a significant shift away from the inherent privacy properties of the DNS, as passing an indication of the original querier's identity to authoritative servers is not a conventional behaviour in the DNS framework. The fact that this Client Subnet option can be attached to a query made by the recursive resolver without the permission of, or even the knowledge of, the original end user is also a problem. Stub resolvers (resolvers used by the end user) may set a source prefix length in a client subnet field passed to a recursive resolver, which is intended to constrain the amount of client address bits that are passed on to the authoritative server. If the Stub resolver uses a source prefix length of 0 in its queries, then the recursive resolver is meant to not use a Client Subnet option in its queries to authoritative servers. However, as with many aspects of the DNS, once a query is passed inward into the DNS infrastructure the precise nature of the handling of the query, and who is privy to the query and the IP identity of the querier is not visible to the end users. The DNS is quite opaque in its mode of operation.

As RFC 7871 notes: "If we were just beginning to design this mechanism, and not documenting existing protocol, it is unlikely that we would have done things exactly this way. … We recommend that the feature be turned off by default in all nameserver software, and that operators only enable it explicitly in those circumstances where it provides a clear benefit for their clients. We also encourage the deployment of means to allow users to make use of the opt-out provided. Finally, we recommend that others avoid techniques that may introduce additional metadata in future work, as it may damage user trust."

There are a number of questions about the use of EDNS Client Subnet, and here we will present some measurements relating to the level of use of Client Subnet in the public Internet.

The overall level of use of Client Subnet is some 12% of users. Its use is concentrated in Africa, parts of Western Asia and South America. A map of the use of Client Subnet is shown in Figure 3.

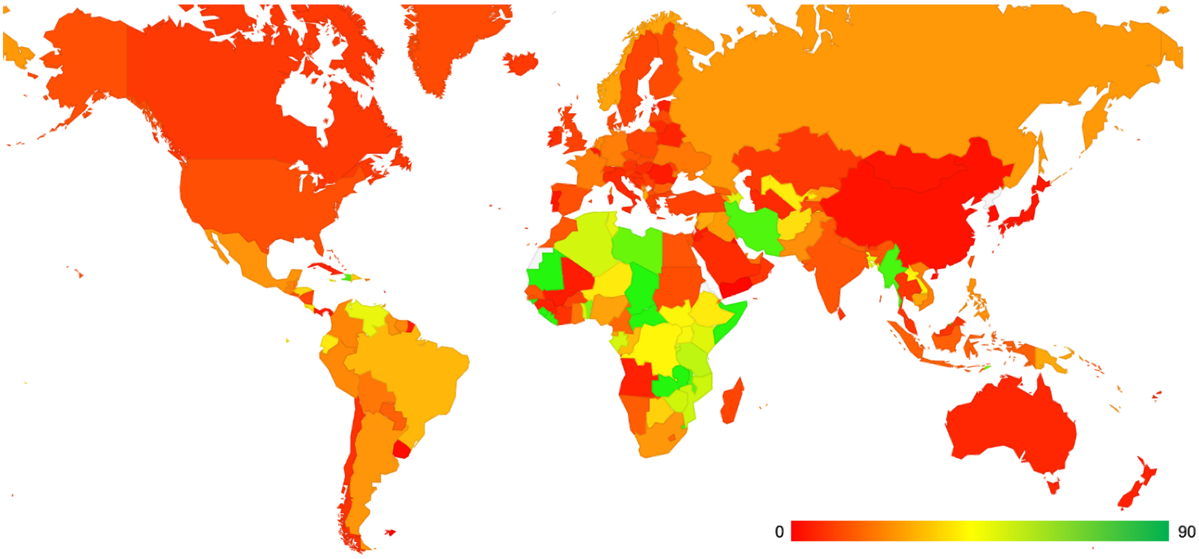

Figure 3 – Use of ECS per country

The ten countries with the highest and lowest levels of use of Client Subnet are shown in Tables 1 and 2.

Rank	CC	Country Name	ECS Query Rate
1	GW	Guinea-Bissau, Western Africa, Africa	100.00%
2	BN	Brunei Darussalam, South-Eastern Asia, Asia	99.60%
3	TD	Chad, Middle Africa, Africa	99.50%
4	CF	Central African Republic, Middle Africa, Africa	99.40%
5	SL	Sierra Leone, Western Africa, Africa	98.90%
6	AD	Andorra, Southern Europe, Europe	97.50%
7	MR	Mauritania, Western Africa, Africa	96.10%
8	ZM	Zambia, Eastern Africa, Africa	93.70%
9	MP	Northern Mariana Islands, Micronesia, Oceania	93.20%
10	SO	Somalia, Eastern Africa, Africa	89.30%

Table 1 – Countries with the highest rate of ECS Use in DNS Queries

Rank	CC	Country Name	ECS Query Rate
1	YE	Yemen, Western Asia, Asia	1.40%
2	UY	Uruguay, South America, Americas	2.30%
3	MQ	Martinique, Caribbean, Americas	2.70%
4	KW	Kuwait, Western Asia, Asia	3.00%
5	GP	Guadeloupe, Caribbean, Americas	3.10%
6	KR	Republic of Korea, Eastern Asia, Asia	3.10%
7	NR	Nauru, Micronesia, Oceania	3.30%
8	CN	China, Eastern Asia, Asia	3.50%
9	TO	Tonga, Polynesia, Oceania	4.20%
10	MN	Mongolia, Eastern Asia, Asia	4.20%

Table 2 – Countries with the Lowest rate of ECS Use in DNS Queries

Client Subnet Prefix Sizes and Match to Client

Some 99% of those users that have client subnet values attached to their DNS queries use a /24 subnet with using an IPv4 network. The comparable common subnet size in IPv6 is a/56 subnet. This distribution of subnet sizes is shown in Table 3.

IPv4 Prefix Size	Count	% Match
/0	21	100%
/20	2	0%
/21	1	0%
/22	39,437	74%
/23	3	0%
/24	63,393,321	80%
/25	275,451	0%
/28	11,801	99%
/32	265,917	0%
TOTAL IPv4	63,985,954	79%

IPv6 Prefix Size	Count	% Match
/0	52	100%
/24	258	0%
/32	76,235	93%
/40	881,495	0%
/48	459,731	45%
/56	141,141,921	67%
/64	4,083	84%
TOTAL IPv6	142,563,775	67%

Table 3 – Client Subnet Prefix Size Distribution

The measurement we are using here is an active measurement, so we are also able to compare the subnet being specified in the DNS queries with the originating client's IP address. The proportion of queries where the Client Subnet value in the DNS query matches the Client's IP address is given in the "% Match" column in Table 3.

In the case of IPv4, the subnet given in the DNS query matches the subnet part of the client's IP address in 79% of cases across all subnet sizes. This number falls to 67% of cases in IPv6.

There are a number of cases where the subnet given is a /32 in IPv4, which is an IPv4 host address of course. This occurred 265,917 times in this measurement (taken in July 2024), out of a total of 63,985,954 tests, so the incidence of these /32 subnets is not very high.

This use of a complete host 32-bit host address rather than a subnet, would appear to defeat even the somewhat limited client privacy measures of this client-side location signalling. However, it's not as bad as these numbers might suggest. Of the 265,917 instances where a host address was used, the address used in the Client Subnet option was not the client's address most of the time. The Client Subnet prefix matched the end client's address just 288 times, or 0.11% of these /32 subnet cases in IPv4. It appears that the resolver is using a location identification framework that is not directly derived from the client's IP address, as a means of mitigating the privacy issues associated with the use of ECS.

In IPv6 no host addresses (a subnet size of 128) were observed, nor any subnet sizes greater than a /64. The most specific subnet size was a /64, and in that case 3,435 out of a total of 4,083 cases (or 84%) of these /64 subnets matched the client end point's IPv6 address. It is not clear to what extent these /64 subnets compromise a client's privacy, but in many cases a /64 is assigned to each client in an IPv6 deployment (https://www.potaroo.net/ispcol/2024-04/ipv6-prefixes.html).

Does ECS Help?

The intent of ECS is to refine the authoritative server's view of the assumed location of the end client by adjusting the DNS steering function performed by the authoritative server from the location of the recursive resolver that generated the DNS query to the location of the client subnet as provided in the query.

The first question here is: Do these two locations (the location of the recursive resolver and the location of the client end point) differ?

If we use a geolocation function that maps IP addresses to country codes, then over a 22 day period in June and July 2024 we saw a total of 104M queries with ECS present and of those queries some 38% geolocated the recursive resolver IP address into the same country as the Client Subnet field. In other words, if the granularity of the geolocation function is at the level of a country, then Client Subnet is offering different location information from that of the recursive resolver in two thirds of observed cases.

What about using a finer level of location granularity, namely that of a match of networks using their Autonomous System (AS) number? Here we map the recursive resolver's IP address to the network that announces this address into the routing system, and compare it to the network that announces the client subnet in the query. Here only 0.57% of cases show the same network AS number for both the resolver and the client subnet. This indicates that ECS is used predominately by non-local network DNS recursive resolvers, which is very much the intended domain of use of ECS.

Another way to look at the effectiveness of ECS is by looking at the three-way relationship between the endpoint's country geolocation and network (Autonomous System Number), the recursive resolver's country and ASN and the ECS values. This is shown as a Venn Diagram in Figure 2. The objective of the ECS signal is to maximise areas B and D of this Venn diagram, where the Client Subnet in the query accurately represents the location of the client end point in terms of both the geo-located country and the origin network AS.

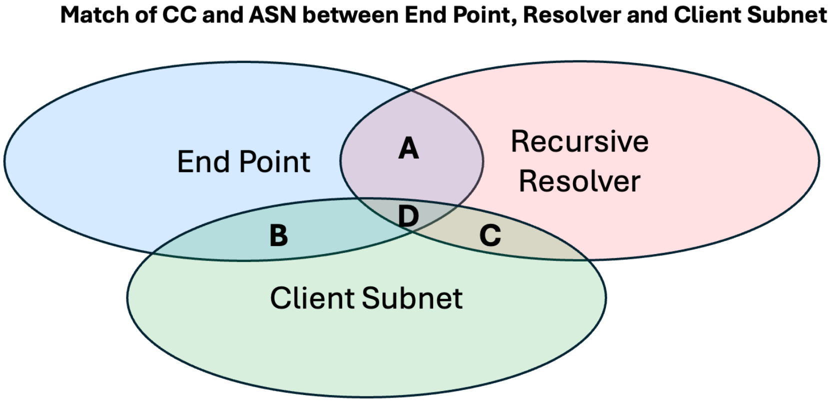
Match of CC and ASN between End Point, Resolver and Client Subnet
End Point
A
Recursive Resolver
D
B
C
Client Subnet

Figure 4 – Venn Diagram of End point, Resolver and Client Subnet

IPv4	A	B	C	D	Remainder
End Point	0.3%	87.8%		0.0%	11.9%
Resolver	0.3%		0.1%	0.0%	99.6%
ECS		**87.8%**	0.1%	0.0%	12.1%

IPv6	A	B	C	D	Remainder
End Point	0.6%	97.3%		0.0%	2.1%
Resolver	0.6%		0.0%	0.0%	99.4%
ECS		**97.3%**	0.0%	0.0%	2.7%

Table 4 – Match of Client End Point, Recursive Resolver and Client Subnet

This is a positive result, in that it indicates that in the majority of cases (88% in IPv4 and 97% in IPv6) the ECS parameters match, in terms of geolocated country and network origin AS, the characteristics of the client end point.

The final question relates to the identity of the DNS resolver that is providing this ECS signal. Table 5 provides these results for the most commonly used open DNS resolvers, as well as the additional categories where the resolver and the client are located in the same AS, or on the same country.

Resolver	Use
Same AS	0.4%
Same CC	0.9%
Other	5.0%
Google	90.1%
Cloudflare	3.6%
Quad9	0.0%

Table 5 – Resolvers who pass Client Subnet in their queries

It is clear that the overall majority of use of ECS is via the Google DNS service. The Cloudflare DNS resolution service is the next most commonly used, although the measurement approach used here cannot discern whether this is through Cloudflare's 1.1.1.1 resolver service or through other DNS resolution services operated by Cloudflare. Our tests indicate that Cloudflare's 1.1.1.1 service does not add ECS to its queries, so the result here is presumably due to some VPN-like service operated by Cloudflare.

These measurements point to ECS largely being used in a manner according to the original intentions of RFC 7871, namely to refine the DNS query with additional metadata to allow an authoritative server to return a response that is the best match between the set of anycast service instances and the location of the end user querying for the service.

Conclusions

With a use rate of some 12% of the Internet's user base, the use of Client Subnet is an integral part of the DNS landscape today, despite the legitimate concerns about the privacy leakage of the end user's IP identity into the recursive-to-authoritative part of the DNS resolution environment, and into the larger DNS infrastructure.

The predominate use of the /24 subnet in IPv4 and the /56 subnet in IPv6 mitigates these privacy concerns to some very small extent, but we could likely achieve the same outcome without referencing the customer's IP address to achieve the same geolocation outcome.

If we had the opportunity to revisit the entire framework, then presumably we might use some other form of location coding that allowed the user to select the level of geographic precision, and trade off the generic issues with location privacy against the quality of the subsequently selected service from the replicated service cloud.

Even if we stick with IP addresses, we can do a better job to minimize the privacy issues. An example of this approach is Adguard's "Privacy Friendly ECS" which uses a randomly selected subnet for each distinct AS, and provide that subnet instead. This may impact on the cache effectiveness of this measure, so they then map all the smaller ASes in a country to the larger networks, improving the cache hit rate without compromising the geolocation function.

There is a tradeoff going on here between service performance and privacy. The more we want the network infrastructure to select the "closest" instance of a replicated service the more we need to be prepared to expose our location to this infrastructure, and the greater the potential for compromise user privacy.

The pragmatic observation is that in today's DNS environment users do not have a lot of control over the level of privacy compromise that occurs in the process of DNS name resolution other than carefully selecting an open recursive resolver that matches their individual privacy preferences.

A report on the level of use of EDNS(0) Client Subnet in queries can be found at https://stats.labs.apnic.net/ecs.

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