Packet Clearing House
572B Ruger Street, Box 29920
The Presidio of San Francisco
San Francisco, California
94129-0920 USA
+1 415 831 3100 main
+1 415 831 3101 fax

Observations on Anycast Topology and Performance

Steve Gibbard
Packet Clearing House
August 6, 2007

Introduction

Many DNS systems are being anycasted. These systems include much of the Internet's critical DNS infrastructure—several of the root servers and many top level domains—as well as private infrastructure for individual or corporate domain hosting. Anycasting of name servers has several advantages, including boosting reliability, increasing resistance to denial of service attacks, and improving performance by bringing servers closer to the users—particularly users in areas far from the Internet's original DNS infrastructure. There are significant variations in network topology between systems, including whether the nodes are completely disconnected from each other or part of a connected backbone, and, if they are disconnected, whether they use a single set of transit providers everywhere or different transit providers in different locations.

There have been several studies of stability and query distribution of anycast networks. Most have focused on a single anycast system and may have told us more about the specific system studied than about anycast in general. A recent study by Ziqian Liu, Bradley Huffaker, Marina Fomenkov, Nevil Brownlee, and kc claffy applied a consistent methodology to three different anycast systems and found significant differences in query distribution among the three.[1]

In this paper I examine the Liu et al. study from a network engineering perspective. I consider how network topology affects performance, as defined by queries going to nearby servers, and why their research showed the results it did. I then look at what we can infer from these results about the behavior of other anycast systems and examine query data for another anycast system to test those inferences.

Liu et al. assume that responding to queries from local servers is desirable, and I make that assumption as well. They use distance as a proxy for performance, since network performance is largely influenced by the speed of light in fiber. In addition, the use of local servers means fewer network segments to break between the query source and the query destination.[2]

This paper does not examine reliability or DOS resistance of anycast networks. Though these are also important design considerations, they are best measured by metrics not included here. Nor do I intend a full overview of measurements of the PCH anycast system, which are included here only to answer a specific question.

Previous research

Lorenzo Colitti, Erik Romijn, Henk Uijterwaal, and Andrei Robachevsky studied the performance of anycast on the K Root server.[3] From a network of probes, mostly in Europe, they examined latency to various K Root instances and concluded that most of their probes were being directed to the closest server. They then simulated the effects of removing various global nodes of the anycast system—anycast nodes reachable from the entire Internet, as opposed to nodes reachable only from networks in a specific region—to evaluate changes of performance. They concluded that the presence of the Delhi node did not make a significant improvement from the perspective of their probes, but that the other K Root global nodes were all useful . They did not look at anycast systems other than K Root.

Liu et al. then studied three root server anycast systems, C Root, F Root, and K Root. Unlike Colitti et al., they looked at actual query sources on the servers and gauged their locations and distance from the servers with a geolocation system. They found that sources of queries to local anycast nodes—nodes set up to handle queries only from networks in a particular region—tended to be local as intended, but that the client distribution of the global nodes differed between systems.

Of C Root's clients, 92% used either the closest C Root server or one nearly closest. In contrast, only 35% of F Root's clients and 29% of K root's clients were served by what Liu et al. called their "optimal instances." Liu et al. consider the C Root result to reflect the fact that all of its nodes are global nodes. That C Root has significantly fewer nodes than the other systems could also explain this result. However, their data showed some significant differences in query distribution among the global nodes of the different systems as well.

The C Root server, operated by Cogent, has four global nodes. All are in the US. The geographic center of the queries to the C Root Chicago node was near Chicago, and it got almost all its traffic from the Americas. The geographic center of queries to the Los Angeles node was in Asia, and its traffic came from a mixture of the Americas, Asia, and Oceania. The geographic centers of the queries to the New York and DC nodes were in the Atlantic, and their traffic came from a mixture of the Americas, Europe, Africa, and Asia (West Asia is generally reached from the US via Europe). This distribution of query sources is optimal, given the locations of the servers. Traffic from Europe, Africa, and West Asia went to the US East Coast nodes because those were the closest available. Traffic from the rest of Asia and Oceania went to the US West Coast because it was closest. Traffic from the central US stayed there.

K Root has five global nodes, in Miami, London, Amsterdam, Delhi, and Tokyo. Of K Root's clients, 29% were served by their "optimal instances." Though the query sources for most nodes were centered near the node, those for the Delhi node were not. Its query sources were centered in North America, from which 60% of its traffic came despite being on the other side of the world. Likewise, the London node got 40% of its traffic from the Americas and 25% from Asia. In contrast, almost all the Miami node traffic came from the Americas, the Tokyo node got all its traffic from Asia and Oceania, and the Amsterdam node traffic was mostly from Europe.

F Root has only two global nodes, in San Francisco and Palo Alto, both of which receive queries from all over the world. Because the two are only 40 miles apart, there is no geographic optimization between them.

Explanation

What explains these differences of query distribution for the C and K roots? The answer lies in the way Internet routing works and in the transit arrangements for the two sets of anycast nodes.

Internet routing follows financial relationships. If a network has traffic to send somewhere, it is best to get paid for sending it, second-best not to have to pay, and worst to have to pay. Using the “local preference” BGP attribute, customer routes tend to be preferred over peer routes, and peer routes tend to be preferred over transit routes. When local preferences are equal, AS path length comes into play, but in this age of global ASes, AS path has little to do with distance.

This is a significant difference between BGP routing, used between autonomous networks, and the various interior gateway protocols (IGPs) used to propagate routing information within a single network. Whereas IGPs are generally designed to send traffic down the shortest or otherwise most optimal path, BGP is designed largely for policy enforcement. BGP’s rather limited metrics for determining path length come late in its selection process, after many routing decisions have already been made.

For end sites in a single developed world location, this distinction does not come to much. “Global” ASes tend to interconnect “globally” (with some exceptions), and more local ASes tend to interconnect within their coverage areas. Hot potato routing—the practice of handing off outbound traffic at the nearest possible exit point—tends to produce reasonably direct paths, no matter which networks the traffic passes through along the way.

With anycast, things are a bit different. If an anycast system gets transit from different networks in different places, it may find its incoming traffic following customer relationships in undesirable ways. If the transit provider for a node in one part of the world is a big global network, or a customer of a big global network, or a customer of a customer, any traffic that hits that big global network in other parts of the world will flow downstream into that node, regardless of the distance.

C Root has all its nodes on Cogent’s backbone, which like most big Internet backbones has a consistent set of peers across its coverage area. Internet traffic destined to Cogent is handed off to Cogent at the closest point of interconnection, because of hot potato routing. Cogent, again using the hot potato system, hands it off to the nearest anycast server.

In contrast, K Root has different transit providers in different locations. When the Liu et al. Delhi node data was collected, the node had transit from STPI, an Indian ISP, which was indirectly a customer of Level3, a global network with a large customer base and extensive peering in North America.[4] Any traffic destined for K Root that entered Level3’s network in the US, sourced from either Level3’s customers or customers of its peers, followed the trail of customer relationships to India. Likewise, the London node is a customer of AboveNet, another US network. Traffic destined to K Root that hits AboveNet’s network is carried to London.

That such an imbalance is not noticeable in the other nodes seems best explained by traffic volumes and the US-centricity of transit arrangements. Though the Miami and Delhi nodes get the highest percentages of their traffic from the Americas, more traffic from the Americas goes to the London node than to those two combined.[5] The Amsterdam node seems to draw less traffic from the Americas than the London node, but it still gets more traffic from the Americas than the Delhi node does. Amsterdam also gets less traffic from Europe than the London node does, suggesting that its efforts to look like a less optimal path may work more broadly than would be desired. Only the Tokyo node avoids drawing significant numbers of queries from outside its region, but, again, the London node gets almost three times as much Asian traffic as the Tokyo node.

Other anycast systems — inferences

Several other anycast systems provide root and top level domain name services, including the J, I, and M Root servers and the name servers operated by UltraDNS that provide name service for .ORG and several other large TLDs. Liu et al. did not study these systems, and I do not have query distribution data for them. We can, however, assume that the characteristics are likely to be similar to other systems sharing the same topology. The J, I, and M Roots all have topologies similar to that of K Root and are expected to have similar behavior.

J Root has global nodes in many different places with many different transit arrangements. In December 2006, traceroutes from three hosts in the San Francisco Bay Area went to J Root nodes in Seoul, Toronto, and Amsterdam, despite the presence of multiple J Root nodes in close proximity.[6]

Though Autonomica, I Root's operator, declines to discuss numbers or locations, they have several full-transit global nodes, on multiple continents, with transit providers "as different as [they]'ve been able to make them."[7] This is an active decision on Autonomica's part, with the intent to increase reliability and security. This policy can be expected to have the same effect of directing queries to faraway servers as experienced by J and K Roots. Autonomica also has several I Root nodes with extensive peering, which should help keep traffic more local than it would otherwise be.

M Root has only two global nodes, but its topology is also similar to K Root and the same behavior is expected.[8] It has transit from several networks in Paris and several others in Tokyo.

UltraDNS operates six different anycast service addresses. Some of them appear to be at least somewhat consistent in their transit arrangements. Traceroutes from several locations using UltraDNS's transit go into geographically proximate nodes: Chicago to Chicago, Phoenix to San Jose, Miami to Ashburn, and Paris to London. Others use a much larger variety of transit, with the accompanying longer distances from query sources to responding servers. In addition to their publicly reachable DNS servers, UltraDNS colocates servers with the caching resolvers of some major ISPs. Only the caching resolvers they are colocated with use them, so they should be very close to their query sources.

Putting inferences to the test: the PCH anycast network

To test the previous conclusions about topology, I ran a similar test on the PCH anycast system, which provides service for sixteen country code top level domains. The global node topology is somewhat similar to that of C Root: in addition to the peering-only local nodes, there are four global nodes spread around the world, all connected to the same two transit providers. The global nodes are in London, Ashburn, Palo Alto, and Hong Kong, and all have transit from both NTT and Teleglobe. In addition, the global nodes have some peering, but with a peering policy requesting that those who peer with them do so with all nodes in areas of overlap. If the preceding topological explanation is accurate, the four global nodes should see traffic only from their rough quadrants of the world.

Local nodes of the PCH anycast system were installed in a mixture of major developed-world Internet hubs and at exchange points in poorly connected parts of the world. For the global nodes (the object of study here), we chose London, Ashburn, Palo Alto, and Hong Kong as locations closest to being evenly spread around the world but still with well-connected Internet hubs with access to major transit. Though we have been careful to select transit providers uniformly—picking two global transit providers for redundancy and refusing transit from anyone else—we have been far more open about peering in our local nodes. Our thinking is that if networks peer with us in locations that we do not consider optimal, or refuse to peer with us where it would be optimal, this only affects performance to their own customers. The flexibility on peering has led to some anomalies in the data, which I address in the following paragraphs.

I tested my topological assumptions by collecting query data over a 24-hour period—roughly midnight to midnight US Pacific time on Sunday, January 28, 2007. Query sources were aggregated by "/24" IP subnet (256 IP addresses) to produce a list of unique /24s that had sent queries to each server. Using RIR whois data, these /24s were sorted by country of registration, producing a count of how many unique /24s from each country had queried each server. /24s were chosen because they are generally the smallest unit of routable address space, so DNS servers in the same /24 can be assumed to be in geographically similar locations. An institutional privacy policy prevented looking at the data on a more granular level. I did not weight sources by query volume, since this was a look at network topology and I wanted an even view. What was produced is thus a rough count of query sources rather than a count of individual queries. I then looked at the percentage of query sources from each country that queried each individual server.

During this 24-hour period, we saw queries from 155,830 /24s. Though the RIR-listed country of registration is unlikely to be the actual location for all addresses, in the aggregate the data should be statistically valid.

The global nodes performed more or less as expected. The servers did not have clearly delineated boundaries between their coverage regions, but only the server in Ashburn saw significant numbers of queries from the most distant parts of the world. Ashburn's case has some reasonable explanations.

The Palo Alto server (Figure 1) answered queries from the Americas, Asia, and Australia. It did not see significant volume from Europe, Africa, or the Middle East. The London server (Figure 2) served Europe, the Middle East, and parts of Africa as well as parts of South Asia; it had no significant volume from the Americas or East Asia. Hong Kong's (Figure 3) query sources were mostly confined to East Asia and Oceania; 2% of its query sources were from Spain, all of which were within Jazz Telecom, a network that peers with us only through the Hong Kong Internet Exchange route server.

Ashburn (Figure 4) seemed most "leaky," seeing substantial numbers of queries from not only the Americas and Western Europe (including 66% of the query sources from Italy) but also parts of Africa, 47% of the query sources from India, and 9% from Japan. This is partly explained by the US East Coast's status as the historical center of the Internet. Many parts of Africa are linked by satellite to the US East Coast, which puts Africa topologically closer to Ashburn than to London. The Japanese queries were all from customers of KDDI and reached us via peering connections rather than via our global transit. Though we peer with them in both Palo Alto and Ashburn, and Palo Alto is closer, they were for some reason violating the usual "best exit" practice by handing traffic off to us in Ashburn instead (in subsequent tests, this appeared to have been corrected). Likewise, more than half of the Italian query sources were within Telecom Italia, which peers with us only in the US. India is more of a mystery. Almost half of the Indian query sources seen in Ashburn were from Bharti Infocom. From our vantage point, they appear to have transit from Deutsche Telecom in London, and traffic on the Deutsche Telecom network in Europe reaches our London node.[9] It is likely that they also have some other source of transit that ends up closer to the US East Coast. Because at least two of the three significant geographical anomalies seen in Ashburn were a result of non-uniform peering arrangements, they are a further demonstration of the importance of uniform peering and transit among global nodes.

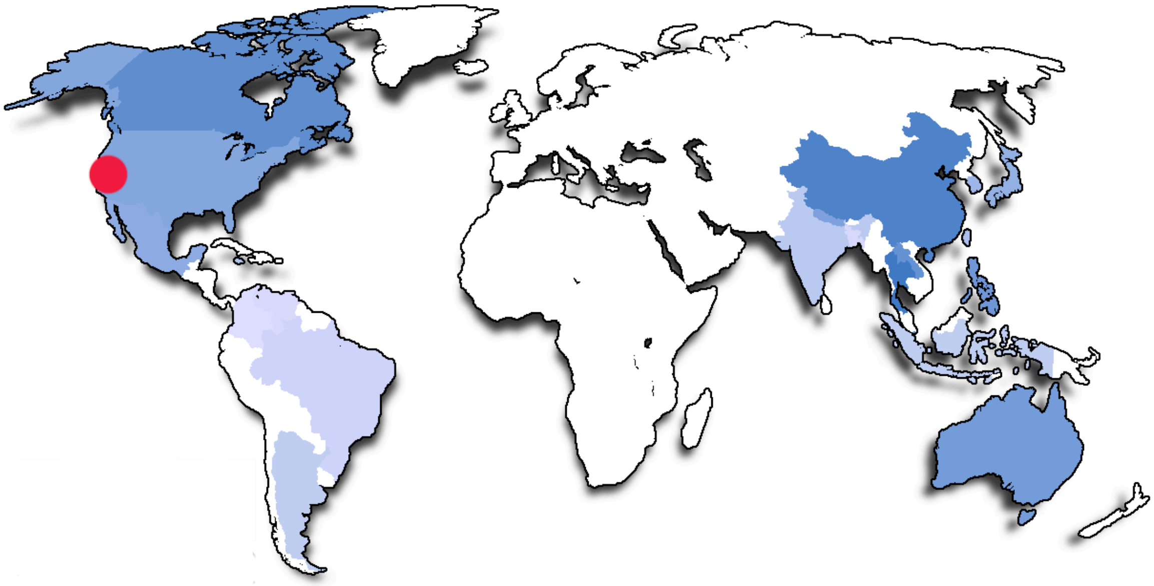

Figure 1: Query distribution for Palo Alto node

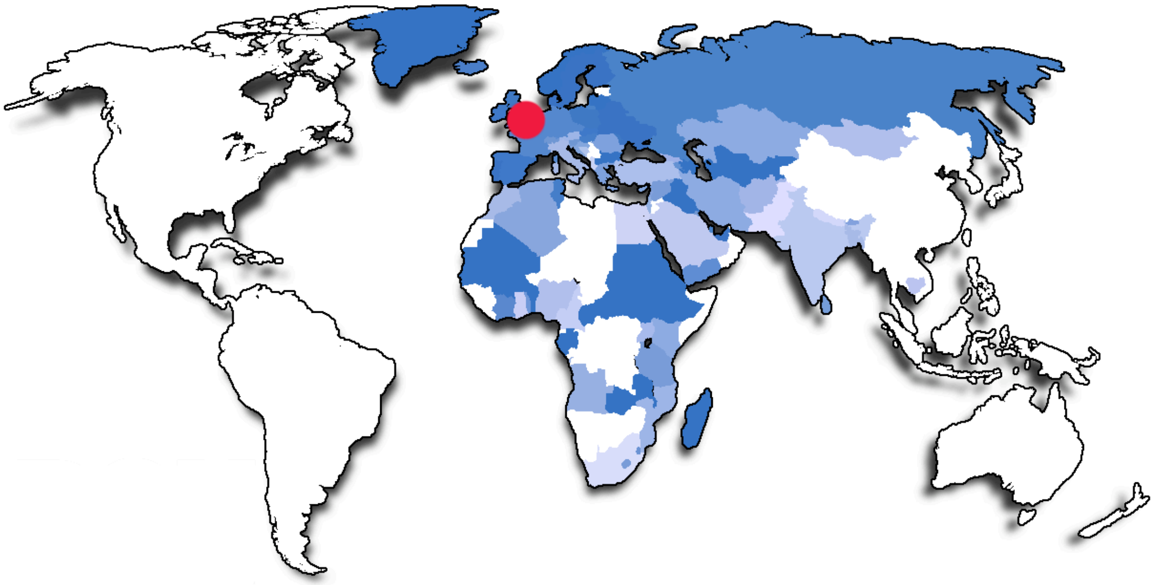

Figure 2: Query distribution for London node

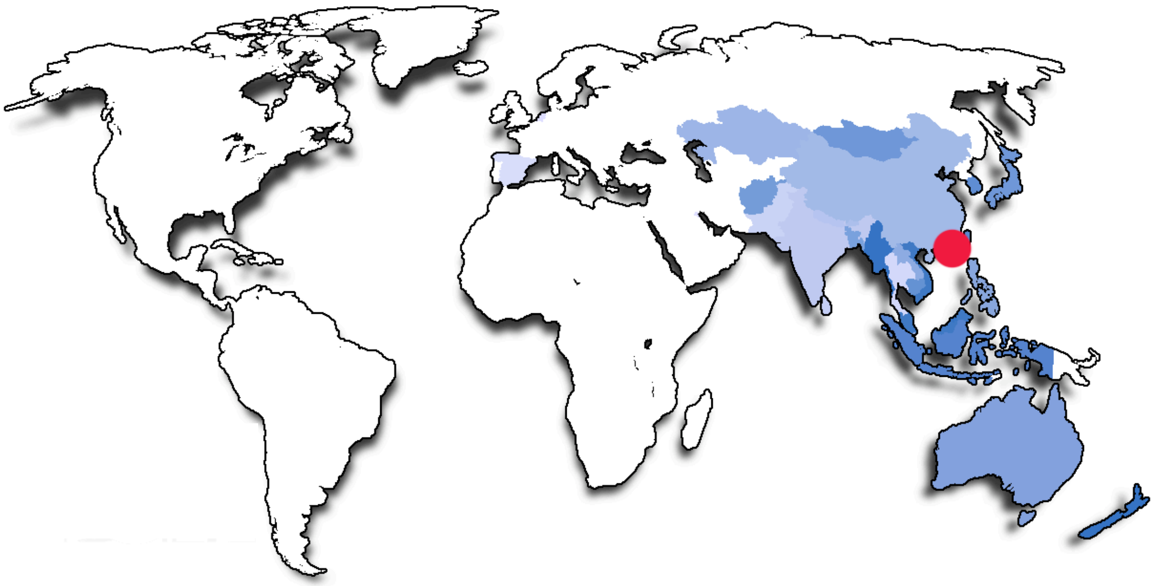

Figure 3: Query distribution for Hong Kong node

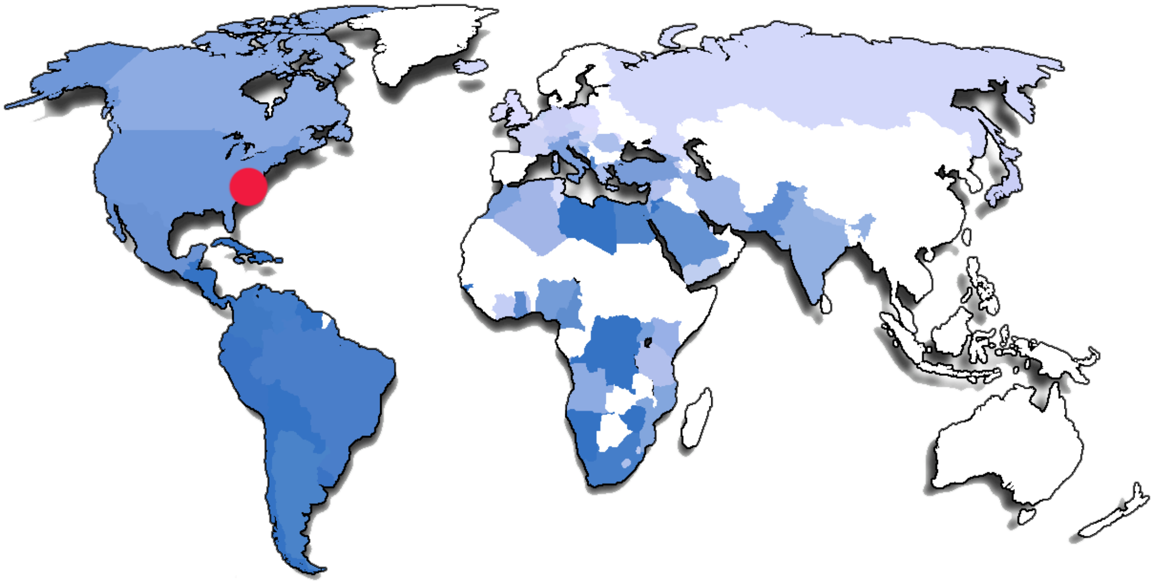

Figure 4: Query distribution for Ashburn node

Conclusions

Designing an anycast system as one would design a backbone network—being consistent about performance and transit arrangements—makes a significant difference in the distribution of query sources among anycast nodes and can thus be expected to make a big difference in terms of performance. Those operators wishing to maintain network diversity by connecting different nodes to different transit providers should nonetheless make an effort to select transit providers whose footprint matches the area the anycast system is intended to serve and to distribute nodes in diverse locations with the network of each transit provider.

Acknowledgments

Thanks to Lorenzo Colitti, Bradley Huffaker, Keith Mitchell, and Bill Woodcock for their helpful comments and suggestions on this paper. Bob Arasmith helped considerably with the data analysis programming, and Vijay Kumar Adhikari did the programming to generate the maps.

This work supported in part by a generous grant from the
Internet Corporation for Assigned Names and Numbers

Notes

[1] Ziqian Liu, Bradley Huffaker, Marina Fomenkov, Nevil Brownlee, and kc claffy, "Two days in the life of DNS root servers," Passive and Active Measurement Conference, 2007.

[2] Steve Gibbard, "Internet Mini Cores: Local communications in the Internet's spur regions," SANOG, February 2005.

[3] L. Colitti, E. Romijn, H. Uijterwaal, and A. Robachevsky, "Evaluating the effects of anycast on DNS root name servers," RIPE document RIPE-393, October 2006.

[4] Route Views Project, data file for January 10, 2006. http://archive.routeviews.org/oix-route-views/2006.01/oix-full-snapshot-2006-01-10-1800.dat.bz2.

[5] Traffic graphs on http://k.root-servers.org, December 18, 2006.

[6] Traceroutes from smalley.gibbard.org, sprockets.gibbard.org, and a host in the 204.61.208.0/21 address block, December 18, 2006.

[7] Johan Ihren, personal correspondence.

[8] Yuji Sekiya, personal correspondence.

[9] Traceroutes to 204.61.216.4 from "London" and "Frankfurt," DTAG looking-glass, https://f-lga1.f.de.net.dtag.de/index.php, January 31, 2007.