Reconsidering Peer-to-peer DNS

John Kristoff
jtk@depaul.edu

ABSTRACT

The domain name system is a hierarchically distributed, and partitioned naming service. Administration, performance, and resiliency relies on a delegated tree-structure of zone authorities, servers, and caching infrastructure. The DNS has become a widely successful, important, and critical subsystem of the larger Internet system itself. Yet, there remains a number of performance, integrity, and availability issues that have periodically threatened the Internet naming infrastructure. A variety of enhancements to the DNS, such as BGP anycast and DNSSEC have arise to address a variety of desires and needs, but so too have attempts to re-imagine and replace the legacy DNS with something significantly different. Thus far, changes and replacements have been met with limited success. One novel approach to a new naming service is to run atop a distributed systems peer-to-peer overlay. Peer-to-peer applications proliferated at the onset of the 21st century with Napster. More recently, blockchain-based systems have arisen with great fanfare and interest. Proposals and experiments for a peer-to-peer DNS have had limited success and are often met with skepticism by today's operators. We survey the literature and offer some modest suggestions to address some of the technical challenges that have plagued past peer-to-peer DNS projects. While we concur with many of the criticisms that have been lodged against a peer-to-peer DNS, we offer some fresh perspectives and a hybrid approach that we believe may help address configuration consistency and denial-of-service threats, while maintaining backwards compatibility with administratively desirable policy capabilities.

ACM Reference format:
John Kristoff. 2016. Reconsidering Peer-to-peer DNS. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 6 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

At the turn of the twenty-first century, with the rise of file-sharing applications such as *Napster* and *Gnutella*, peer-to-peer networking overlays stirred the imagination of Internet users, operators, and researchers. By moving, limiting, or eliminating hierarchy and control, file sharing applications highlighted advantages a peer-to-peer overlay network might offer other critical services and applications. One such peer-to-peer service that has been subject to attempts of a reimagining as a peer-to-peer service is the domain name system (DNS).

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

In this paper we provide a brief introduction to the DNS, summarizing its important architectural design properties. This background will prove helpful in understanding Internet DNS challenges peer-to-peer alternatives have sought to address. We then survey and review the history of peer-to-peer DNS systems and ideas in the available literature. We highlight the key contributions and innovations as well as some of their limitations. We conclude by offering some new ideas for the DNS in the areas of delegation management, performance, policy enforcement, and debugging capabilities.

2 THE DOMAIN NAME SYSTEM (DNS)

The Internet domain name system (DNS) may be the Internet's largest and most important distributed systems service with perhaps the sole exception of IP routing.[10][11] The DNS provides a mapping of names to data about those names, most famously host name mappings to IP addresses. The DNS is made up of two basic complimentary functions, one is a process to make data about names available and another process to retrieve the data for those names of interest. The former is more commonly referred to as an authoritative service and the later as a resolver service. For scaling and security, these two services are usually decoupled and run independently, but they need not be.

In theory the entire name space could be located in and served by a single authoritative server system. However, for policy, scaling, performance, and availability, the name space is partitioned as a tree with a well agreed upon, known root.

The Internet name space is made up labels separated by dots, with the root anchored at the right-most label and more specific labels added from right to left. For example, in the domain name *www.uic.edu.* the root is the right-most dot, although this is often omitted in practice, *edu* is the least specific label also known as a top-level domain, and the *www* label is the most specific portion of the name.

The name space is partitioned, often on label boundaries, in order to distribute administrative control and authoritative service responsibilities. An authoritative server provides service for one or more parts of the DNS name space tree. When a delegation occurs, access to a branch is indicated by special DNS infrastructure records called name server (NS) resource record sets (RRsets). These are special domain names that indicate the authoritative name servers for the branch being delegated. Parent and child name servers need not have any particular relation other than agreeing that the child is the authoritative source of data at the new tree branch. This tree partitioning and authoritative service delegation system enables great flexibility and scale. It also implies a fair amount of loosely coordinated fate sharing from the root on down the tree, which poses some interesting challenges, some of which peer-to-peer DNS proposals have tried to improve upon.

3 LITERATURE SURVEY

3.1 Chord

The first known documented exploration of peer-to-peer DNS ideas was born out of the distributed hash table work at MIT in the early 20th century. Chord was a distributed hash table (DHT) routing protocol for one of the first formally defined peer-to-peer architectures.[16] Chord maps a unique hash table entry to an IP address in a distributed network node. This routing overlay formed the basis for a simple key/value store that was distributed among equal participating peer-to-peer nodes in the Chord system.

The reader is referred to the original Chord paper and [8] for a full evaluation of Chord capabilities, complexity, and operation. In the context of DNS, we must make a few pertinent notes worth highlighting. Since DNS is often used to locate most application services, at least at start up, DNS latency is of utmost concern. Authoritative and resolver services are often located strategically throughout the network so that data can be reached quickly and cached locally. A Chord ring does not take the underlying network topology into consideration. The consistent and fair hashing algorithm employed by Chord may place keys any where in the network, including at the furthest location from a source making a request. Even if the DNS name space were flattened to match Chord's flat hash-based routing, there is a high probability that popular keys are not going to be located nearby, increasing access time and latency.

The Chord authors anticipated some of these performance limitations and suggested it is up to a higher layer application running on top of Chord to perform replication and caching. Geo-location, while not specifically mentioned may well be another possible application-level optimization worthy of exploration.

The Chord paper was the seminal work for peer-to-peer systems generally, but also for DNS specifically. The authors dedicate a paragraph suggesting a Chord-based DNS service as one of their example use cases. Without hierarchy and with each Chord node able to perform name resolution equally or able to store data for a name in rough equal proportion to the entire network, it was suggested that Chord could eliminate special servers (e.g. root servers) and ease configuration management through the automatically maintained Chord routing overlay. In fact, members of the MIT group soon went on to publish an evaluation of just such a Chord-based DNS which we summarize in the next section.

3.2 DDNS

In follow up work to Chord, Cox et al. published results of efforts to implement a Chord-based DNS using DHash, a distributed hash table built on top of Chord.[2] If Chord maps a key to a node, DHash is the solution to store data into the distributed hash table. It is by placing domain name records in DHash that DDNS (presumably an abbreviation for DHash DNS) was born. DDNS appears to be the first known attempt to consider and evaluate a peer-to-peer DNS.

The authors use a prior study's DNS measurement data to populate a DDNS test bed network where they then proceed to issue queries into the Chord network to measure behavior.[7] While storage of names was roughly balanced when placed randomly, having popular names coincidentally served from a single node was a concern. Since the hashing algorithm is unaware of potential load or co-located name popularity, the system is reliant on the statistical probability that popular names will be fairly evenly distributed. As a result of a caching layer implemented by the DDNS test bed, the authors found popular names would quickly propagate to other Chord nodes as requests grew.

While caching improved data replication throughout the network and therefore overall performance, average latency was measurably worse under DDNS than compared to the measurements found in [7]. Chord maintains and uses approximately $O(\log N)$ messages per resolution on average. As the Chord network grows, the additional latency between nodes as the search algorithm makes its way around the ring adds up. Traditional DNS provides a comparatively simpler and and faster tree topology, where caching isn't just for a specific name, but for entire branches in the tree minimizing significant routing delay. Legacy DNS answers were found to be returned without having to incur a referral (node routing hop) 80% of the time. While popular names obtained reasonable good performance in the Chord network, comparing the entire data set under study, the Chord network performed almost an order of magnitude worse than legacy DNS![1]

Despite the performance drawbacks however, DDNS highlighted two areas where a peer-to-peer DNS show some potential advantage. One is the area of denial-of-service (DoS) resistance. While key delegation points in legacy DNS are susceptible to coordinated DoS attacks, a peer-to-peer overlay mitigates this problem by placing no particular importance on any particular node on the ring. Therefore, the larger number and greater diversity of nodes on the ring, the more resistant to a directed attack the network becomes. Another less appreciated, but perhaps no less important advantage a peer-to-peer DNS offers is the ability to eliminate some of the manual delegation effort that often leads to configuration errors in the legacy DNS. Peer-to-peer DNS routing configuration is eased because delegation records no longer need to be manually coordinated at branches, with parents and children cooperating to maintain consistent NS RRsets. The Chord ring is self-forming and handles routing automatically rather than having to be manually configured at each delegation point in the legacy DNS name space. The authors highlighted both the potential advantages and drawbacks of a peer-to-peer DNS system. They seemed to conclude pessimistically that a peer-to-peer DNS was inferior to the legacy hierarchical tree-based DNS. Yet, still more attempts to implement a naming service using a peer-to-peer overlay would continue to attract researchers and experiments as we outline throughout the remainder of this section.

3.3 Overlook

Overlook is a peer-to-peer name service proposal not unlike DDNS, but designed to be a general-purpose naming service not specifically a replacement to the legacy DNS.[17] Unlike DDNS, Overlook uses a Pastry-based peer-to-peer network instead of Chord for its DHT routing overlay.[14] Like DDNS, Overlook actively caches replica data as requests are relayed through routed nodes. However, Overlook actively manages caching for the purposes of later proactively pushing updates to those replica nodes. While this optimization may decrease the average latency compared to DDNS

[1] 350 millisecond average response time versus 43 millisecond average response time.

in some cases, particularly for highly volatile name space entries, Overlook also suffers high latency overhead on average compared to the legacy DNS. Overlook performs roughly $O(\log N)$ look ups as other DHT-based systems do.

Perhaps one of the key architectural differences in Overlook is less theoretical and more of a practical one. The authors opt for a limited set of peer nodes that they control and deploy rather than making the Pastry overlay open to all. This limits the variability in node capabilities and helps to ensure some minimal standard of performance, as well as some amount of administrative control over the network. They reason that there is also little incentive for third parties to adequately provision and maintain Overlook peer nodes that help support an infrastructure they may or may not directly benefit from. This separation of a peer-to-peer backbone infrastructure network enables stricter control of name look ups into Overlook, allowing for some degree of admission control and authentication if desired.

3.4 PNRP

At the turn of the century, Christian Huitema, already well known in the Internet community, having served on the Internet Architecture Board (IAB) from 1991 to 1996 joined Microsoft Corporation as an architect within their networking and service division. Christian spent much of his time providing guidance on a number of Internet technologies including IPv6, SIP, NAT traversal, and peer-to-peer overlay systems. As part of the peer-to-peer effort, he was a key member of a team that spearheaded the Peer Name Resolution Protocol (PNRP) that was added to Windows XP and remains an integral part of the Windows operating system to this day.[5] PNRP is a promising commercial implementation of a peer-to-peer overlay with widespread deployment on Microsoft platforms. That it is widely available for Microsoft operating systems probably makes it the most widely available and deployed peer-to-peer system software to date.

The PNRP however has been less well studied in academic literature, although the specification is fairly well documented in a regularly updated specification published by Microsoft's development network library.[9] PNRP routing is similar to Chord and Pastry. Many of the differences are largely implementation-specific that include optimization for easing inter-connectivity between ad hoc distributed Microsoft platform nodes. The specification dictates that a node ID is to be based on an IPv6 address. This may have limited initial deployment since at the time, IPv6 connectivity was not widely available for most users. Until most end users obtain IPv6 connectivity outside of their local area network a large-scale PNRP infrastructure may be just beyond their reach.

While PNRP seems relegated to a LAN-based service discovery and ad hoc networking mechanism in Microsoft Windows today, that this technology is widely available and deployed on existing systems would seem to make it a convenient starting point for future peer-to-peer overlay experiments and research.

3.5 CoDoNS

In 2004, with little fanfare, another proposal for a peer-to-peer DNS was published in the ACM SIGCOMM Computer Communications Review magazine.[13] CoDoNS also used a DHT-based overlay as the basis for routing, but added an aggressive caching layer that could be controlled to reduce peer-to-peer routing hops to $O(1)$, which would effectively replicate all data to all nodes and promise practically ideal performance. CoDoNS also implemented an early form of *pre-fetching*. When cached data was set to expire, the system would proactively retrieve a fresh copy of the name's data so it would continue to be available in the local cache.

CoDoNS wasn't too dissimilar to earlier DHT-based routing systems overlaid with a DNS application on top. What made CoDoNS particularly noteworthy was how it was ultimately introduced to the Internet community and people's reaction to it. Beginning at the RIPE 52 meeting in Istanbul, Turkey, the lead professor for the CoDoNS paper presented on the topic of transitive trust issues with the legacy DNS. He highlighted how the dependency graphs of the name server delegations were often subtly complex and each server in the graph potentially pose a unique and perhaps unrealized danger to names it may be relied upon to resolve. Discussion outside of the RIPE meeting first went public on Slashdot and a story based on the transitive trust work eventually found its way into a BBC News article with the provocative title "Big holes in the net's heart".[1][19]

The RIPE meeting, Slashdot, and BBC News coverage eventually spilled over to the DNS operations mailing list where many of DNS luminaries can be found. As a discussion thread about the transitive trust and CoDoNS research got going, many well known DNS personalities weighed in and eventually the lead professor from the CoDoNS joined in to respond to criticisms of the work.[18][2] The tone of the discussion became noticeably tense and confrontational. Many in the DNS operations community felt that the transitive trust and CoDoNS paper were over hyped and poorly reflected reality. While many agreed that the research work highlighted concerns worthy of discussion and consideration, peer-to-peer DNS proposals have been largely dismissed and ignored by the DNS operations community ever since. When asked to comment on what they would do differently, Emin Gun Sirer, co-researcher and outspoken representative of the CoDoNS work, gave this partial response:

> "If I had to do everything again, I'd spend much more time appeasing some of the critical personalities in the space and winning them over one-by-one in private. They felt like they were made to look bad by our announcement that most DNS infrastructure was set up in an insecure fashion."[3]

Valid technical critiques aside, sometimes acceptance and deployment problems are as much political as they are technical.

What CoDoNS and other DHT-based peer-to-peer overlay networks struggle with when trying to replace DNS is that while addressing existing concerns such as availability or configuration complexity, a pure peer-to-peer system is not necessarily a better solution in other ways. For instance, with a peer-to-peer system such as CoDoNS, the system becomes much less transparent and predictable. Existing peer-to-peer systems pay little regard to data policy and custom answer synthesis for instance. Many of the features of today's DNS are possible, because the data owner is not

[2]The author of this paper was one of the participants in this discussion. Not one of the luminaries.

only in charge of putting data into the system, but also has some control over who gets access to it and what the results of a look up may be. In many cases the data owner wants to present different responses to different requests. None of the proposed peer-to-peer systems currently provide a mechanism to address these issues of control. In fact, many of them actively seek to undermine it. Since CoDoNS, most peer-to-peer DNS proposals and derivatives seek to not only undermine the ICANN-guided tree hierarchy, but to eliminate most of the centralized control over the name space. It is not clear this is a universally desirable goal, at least not as a total replacement for the Internet DNS.

3.6 DNS Design with DHT-Base Alternatives

The last formal paper in this literature survey is a comparative study of DHT-Based DNS designs.[12] This paper provides a convenient and relevant summary of the performance and availability characteristics of the DHT-based peer-to-peer DNS designs, some of which we have already summarized along more broad lines. This paper provides much in the way of comparison to the legacy DNS that peer-to-peer based systems either failed to provide or did not cover as deeply. For instance, here we discover that the average node degree in the legacy DNS is approximately only 2.5 hops, which compares very favorably with all of the DHT-based systems we have reviewed. DHT-based network path lengths are typically on the order of (1/2) log *N*, which for a peer-to-peer network of comparable size to the legacy DNS is likely to be significantly larger than 2.5.

It can be shown that a DHT-based routing overlay however is sufficiently more resilient to coordinated attacks than most legacy DNS deployments. However, this study did not take into account the widespread use of shared unicast addressing *(aka anycast)* that has become widely used for scaling and availability in the modern DNS.[4] While deployment of anycast is deployed only on certain portions of the name space, it is widely regarded as an effective tool for achieving load balancing and as a low-cost mechanism that helps mitigate packet-flooding style denial-of-service attacks.

This study also considers the effectiveness of caching in both DHT-based peer-to-peer overlays and the legacy DNS. It notes that the effectiveness of the caching strategy can have a significant impact on overall performance. The caching strategy employed not only matters, but the characteristics of the traffic itself will ultimately be reflected in how well the system performs. For instance, a cache is only useful when a name is popular no matter the path length in a peer-to-peer system or the legacy DNS. We can conclude therefore that the data and studies must be representative of real-world name space usage patterns. As such, the claim by CoDoNS that it could achieve O(*1*) hop count latency may be theoretically true, but practically very unlikely.

3.7 Ethereum

To round out this section we consider a more recent development in the peer-to-peer networking space, a blockchain-based system called Ethereum Naming Service (ENS).[6] DHT-based overlay networks, particularly for replacing the DNS have fallen out of fallen favor not long after earlier experiments and studies highlighted the high latency costs, absence of predictability, and lack of centralized policy control (ironically this last attribute some would claim is a feature). With the advent of digital currency, most notably through Bitcoin, cryptographically-based blockchain technology has arose to take the spotlight and attention from the earliest of the decentralized peer-to-peer systems. While blockchain technologies often imply a peer-to-peer overlay network, the essence of a blockchain is the application of an open ledger system on top of a distributed, usually decentralized, system.

A typical blockchain consists of a hash, a timestamp, and a pointer a previous block. Ethereum is one specific and increasingly popular implementation of blockchain technology. Undeniably blockchain technology has garnered significant interest in the past few years. The notation in blockchains are long strings of hashes representing the pointers to the previous blocks. Referring to these hashes are impractical outside of automated processes and software interfaces, since these long strings of characters are cumbersome for humans to memorize and recall.

Perhaps it was inevitable that a blockchain-based naming service would be proposed and implemented. For now, ENS is primarily concerned with mapping names to Ethereum data, as opposed to replacing the legacy DNS. While ENS may not replace the legacy DNS, it appears poised to lead the way in how a naming system application may ultimately be used in decentralized distributed systems. ENS is currently in its earliest phases, but has already began to offer a name registration service through an auction process. Few from the Internet DNS community are activity monitoring or participating in blockchain-based naming systems, but this may be an area to watch how a new naming service evolves on a decentralized distributed system.

4 OUR CONTRIBUTIONS

As we have seen, a peer-to-peer based DNS running atop a DHT-based routing overlay, while intriguing, has thus far failed to supplant the legacy DNS in both practice and in promise. While peer-to-peer based systems present significant advantages in certain domains, such as availability, they pose enormous challenges in others including performance and policy provisioning. Can peer-to-peer DNS be reconsidered such that these disadvantages might be eliminated? Or perhaps additional advantages could be realized by new thinking? We offer the following ideas that when considering the legacy DNS with a mind towards new directions in peer-to-peer overlays.

4.1 Delegation Protocol

One of the major shortcomings in the legacy DNS system is the error-prone nature of name space delegation. As the name space is partitioned and branches are delegated to subservient name owners and name server operators, the bulk of this provisioning process is manually performed. Once configured there are mechanisms, such as the zone transfer, to perform replication for zone data consistency, but the delegation configuration itself cannot be easily automated. A new algorithm and protocol that handles the delegation function and the parent/child name server configuration is considered. Leveraging a DHT such as those outlined in DDNS, we envision a management plane protocol and algorithm forming

small and targeted peer-to-peer networks of parent and child name servers.

We wish to maintain the parent/child hierarchy of the legacy DNS system, since we wish to retain the benefits of a tree hierarchy. This hierarchy best enables fast and efficient routing through the name space by being able to cache entire branches and quickly route to specific branches when aggregate parent branches can be cached, saving invaluable time from traversing the network, thereby reducing the overall resolution hop latency time.

A child and parent branch will form an authenticated private peer-to-peer overlay for the sole purpose of exchanging and updating delegation information. The parent must first be configured with a child's public key. A child should also obtain a parent's key in order to verify they are talking to the expected parent, but this is optional, since updates to a fake parent will have little impact since resolutions would not typically know to contact a fake parent for resolution to start with. Once the parent has been configured with the child's key, a child may configure any child name server with that key and join the parent/child DHT-based peer-to-peer overlay network. This is a private peer-to-peer network that cannot be joined without authentication using the proper child key.

With the advent of conflict-free replicated data types (CRDTs), we could even envision leverage the delegation data as a commutative record type that can be replicated between child and parents in a peer-to-peer topology.[15] This delegation management protocol would ease branch configuration and remove unnecessary misconfigurations due to manual entry lapses.

4.2 Location-based Service Identifier

Simple DHT-based peer-to-peer overlays suffer from too simple routing and replication strategies. Most schemes organize the ring based on random hash values for node ids. While the node ids are often based on the IP address, any IP routing or otherwise location-based identifier is removed in the peer-to-peer overlay. The absence of any location-based identifier means that data is likely going to be stored randomly, and often remotely from where it may be most desired. We propose we add options to store the data based on specific IP address or specified IP address prefix. Additionally, we would offer additional location-based identifiers not typically found in routing systems to help offer additional locality based storage of data on the peer-to-peer network. GPS coordinates, or latency measurements to beacon nodes for instance may help direct where data in the peer-to-peer network may best be located.

Like the aggressive caching strategies performed by DDNS and CoDoNS we would supplement the DHT data with location-aware policy rules and as well as add location-based information to node IDs so that look ups could optimize searches based on location along with the name.

We offer two alternative approaches to location-based routing. One is a geographical based on GPS coordinates. A look up could route based on the closest GPS coordinate node id. Another is a client that could specify a maximum distance threshold, with some reasonable default based on the average hop-count look up depth. Since the average hop count look up depth is O(log N), our scheme would use a distance or regional location than prefers a value equivalent to m away in our Chord finger table for instance.

4.3 Policy Control

A popular DNS feature offered in server software and by commercial DNS providers is the ability to tailor answers based on a number of attributes of the requester. For instance, given a source IP address S_i an authoritative provider may provide tailored response R_i. We wish to provide feature parity in a revised peer-to-peer based system. A peer node answering could tailor the response, but this would require data to contain more than just an answer, but a complex set of instructions for a peer node to compute or relay to the originating requester.

Solving the policy problem in a peer-to-peer network appears harder, especially since we may not want to even divulge policy to just any node in the peer-to-peer overlay.

This may be a situation where nodes in a peer-to-peer network do not necessary have or are able to obtain a consistent view of the data available. If we controlled or dictated the peer nodes we wish to share our data with, we can simply permit or refuse to provide data to those peers based on policy. We envision a mechanism akin to a BGP routing policy where we define *export* and *import* policies based on the peer node we are communicating with. If a peer node meets a particular policy we provide the tailored response, otherwise we provide a negative acknowledgment.

When a peer-to-peer network implements such policy, it suddenly violates some basic assumptions of all nodes being more or less equal. We argue that nodes are already capable of inequality inherently. Some nodes may be unequal due to competing lookup demands, others might be unequal due to policy control. Each case can be treated as a failure, but not one that prevents the system from making or servicing other permitted requests.

4.4 Debugging and Troubleshooting

Our final recommendation is the development of peer-to-peer overlay debugging, interrogation, and monitoring capabilities. Most of the peer-to-peer overlay literature makes little to no mention of the importance of troubleshooting and debugging capabilities. We need at least two capabilities. One is the ability to discover the peer-to-peer topology and the other is to issue look ups that provide a trace of the path and measure of process activity at each node.

Discovery the topology in a peer-to-peer network must be possible in a scale-able way. We would not want just any and every node to discover every single node in a large peer-to-peer network recursively since that may likely open the network to large denial of service attacks. Instead we provide a hop limit mechanism that allows a node to query any node for its successor or finger table.

Even any discovery may pose potential security threats since some nodes may wish to remain anonymous so by default we would recommend anonymous queries be both rate limited and subject to potential access control. A node may wish to limit what other nodes can ask it to do. See the section above regarding policy for implications of these access controls.

A peer-to-peer network should provide the equivalent of a traceroute for IP path discovery, but here for peer-to-peer hop traversal discovery. Our peer-to-peer traceroute tool would work much like traditional traceroute. For each look up we would return the round and peer node identifier plus a latency measurement. In order to

minimize attacks on the peer overlay, forcing peer nodes to incur the work of tracing the path from a search request, each node must provide the data in response or the appropriate successor node it would use to locate the data. It must be up the original lookup source to perform the work iteratively so as not to burden peer nodes with unnecessary effort and network state. This approach would minimize resource exhaustion attacks.

5 CONCLUSION

In this paper we provided evaluated peer-to-peer DNS research literature and attempted to reconsider many of those ideas in light of the modern era. While peer-to-peer DNS never became widely deployed it highlighted important limitations or challenges with the legacy DNS. Of all the promised advantages of a peer-to-peer overlay, the most desirable is the ability to achieve high availability by equally distributing DNS data and load. However replicating DNS data more or less equally comes with significant drawbacks. For a peer-to-peer network to be resilient, it might be of sufficient size and diversity. However, with larger peer-to-peer networks, the latency due to search look up in a flat routing system becomes increasingly undesirable compared to the existing hierarchical, cached approach. The Overlook design suggested a peer-to-peer overlay that is limited in scope and size, to backbone infrastructure instead of to all possibility connected networks and nodes. This hybrid and similar limited approaches are preferable, but offer limited advantage over existing designs. We could envision a separation of the management traffic from the query traffic, where the delegation configuration might be managed by a peer-to-peer protocol, where availability is more important than latency. This is one area in particular that may be worthy of further investigation.

With the advent of blockchain technology we have seen a renewed interest in decentralized naming systems. It remains to be seen whether these will proliferate and find success outside a few niche areas however. The Ehtereum system with ENS is a development worth watching. We however do not expect it or anything like it to supplant the legacy DNS design any time soon. The legacy DNS is not only one of the largest subsystems in the larger Internet systems, but is also one of the oldest. This means it has about as large an installed base as any service on the Internet. We should never underestimate the resistance to change the installed base poses.

REFERENCES

[1] CmdrTaco. 2006. Perils of DNS at RIPE-52. (April 2006). https://slashdot.org/story/06/04/26/1247240/perils-of-dns-at-ripe-52

[2] Frank Dabek, Emma Brunskill, M Frans Kaashoek, David Karger, Robert Morris, Ion Stoica, and Hari Balakrishnan. 2001. Building peer-to-peer systems with Chord, a distributed lookup service. In *Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on*. IEEE, 81–86.

[3] Emin Gun Sirer. 2017. CoDoNS. (Nov. 2017). Personal email communication.

[4] T Hardie. 2002. *Distributing Authoritative Name Servers via Shared Unicast Addresses*. Technical Report. IETF. https://www.rfc-editor.org/info/rfc3258 DOI: 10.17487/RFC3258.

[5] C. Huitema and J.L. Miller. 2006. *Peer-to-peer name resolution protocol (PNRP) and multilevel cache for use therewith*. Google Patents. https://www.google.com/patents/US7065587

[6] Eric Johnson. 2017. ENS Documentation. (Sept. 2017). https://media.readthedocs.org/pdf/ens/latest/ens.pdf

[7] Jaeyeon Jung, Emil Sit, Hari Balakrishnan, and Robert Morris. 2002. DNS performance and the effectiveness of caching. *IEEE/ACM Transactions on networking* 10, 5 (2002), 589–603.

[8] Ajay D Kshemkalyani and Mukesh Singhal. 2011. *Distributed computing: principles, algorithms, and systems*. Cambridge University Press.

[9] Microsoft. 2017. Peer Name Resolution Protocol (PNRP) Version 4.0. (Sept. 2017). https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-PNRP/[MS-PNRP].pdf

[10] Paul Mockapetris. 1987. *Domain Names - Concepts and facilities*. Technical Report. IETF. https://www.rfc-editor.org/info/rfc1304 DOI: 10.17487/RFC1304.

[11] Paul Mockapetris. 1987. *Domain names - implementation and specification*. Technical Report. IETF. https://www.rfc-editor.org/info/rfc1305 DOI: 10.17487/RFC1035.

[12] Vasileios Pappas, Daniel Massey, Andreas Terzis, and Lixia Zhang. 2006. A comparative study of the DNS design with DHT-based alternatives. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*. IEEE, 1–13.

[13] Venugopalan Ramasubramanian and Emin Gn Sirer. 2004. The design and implementation of a next generation name service for the internet. In *ACM SIGCOMM Computer Communication Review*, Vol. 34. ACM, 331–342.

[14] Antony Rowstron and Peter Druschel. 2001. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, 329–350.

[15] Marc Shapiro, Nuno Preguia, Carlos Baquero, and Marek Zawirski. 2011. Conflict-free replicated data types. In *Symposium on Self-Stabilizing Systems*. Springer, 386–400.

[16] Ion Stoica, Robert Morris, David Karger, M Frans Kaashoek, and Hari Balakrishnan. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review* 31, 4 (2001), 149–160.

[17] Marvin Theimer and Michael B Jones. 2002. Overlook: Scalable name service on an overlay network. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*. IEEE, 52–61.

[18] Various. 2006. The dns-operations April 2006 Archive by thread. (April 2006). https://lists.dns-oarc.net/pipermail/dns-operations/2006-April/

[19] Mark Ward. 2006. Big holes in net's heart revealed. *BBC NEWS* (April 2006). http://news.bbc.co.uk/2/hi/technology/4954208.stm