P^{3}DNS

Brendan Benshoof Andrew Rosen
Department of Computer Science, Georgia State University
34 Peachtree St NW
Atlanta, Georgia 30303
bbenshoof@cs.gsu.edu

***Abstract*—We created a software package to replace the top level of the Domain Name system. It is reverse compatible with traditional DNS and requires no modifications on the end of the user.**

I. Introduction

The Domain Name System, commonly referred to as DNS [7] [8], is a fundamental component of the Internet. DNs maps easily remembered names to the numerical IP addresses used by computers to communicate over IP.

There are some issues with DNS, which is pretty much the same now as it was when it was proposed. Now we've inherited those issues. DNS architecture serves the world, but is based in the united states

Two recent events in the United states have brought DNS to the forfront of networking and security research. First is recent legislation proposed in the US House of Representatives [12] and US Senate [10].

The most pressing issues are: An ill advised law that doesn't understand how the internet works. The big trusted certificate issuers are native to the US and don't want get on the bad side of DHS/NSA/FBI/ETC

We have a system that can eliminate or at least lessen the severity of either issue.

This paper is intended to address concerns raised by Cox et al[3] and propose a viable decentralized DNS replacement based on a DHT. Our proposed improvements on the DNS alternative presented by Cox et al are full reverse compatibility with the current hierarchical DNS system and a shared authenticated public record that allows for DNSSEC style authentication.

II. Background

ITT we talk about DNS and work related to this paper.

A. DNS Overview

DNS has a heirarchical structure.

Knocking out certain servers can deny interent access to an entire region or zone.

SOPA [12] is bad [6].

PROTECT IP is bad [4]

These bills would have required that servers maintained in the US filter specified domain names, preventing users from obtaining the correct IP address for the domain name in question. The

This filtering is incompatable with the DNS Securtity Extensions (DNSSEC)[4].

The mandated dns filtering would drive users to unregulated DNS servers, which would create more attack vectors where users could have their traffic rredirected to a malicious website.

B. Related Work

What were the concerns of the cox paper [3]? Higher latency because chord (plus none of the chord advantages). Wasn't extensible. Served static records. Really nothing more than a distributed text file Not incentivised. No reason not be a defector in the system.

Advantages: DDNS is more resistant to DDOS attacks. Load balancing

III. Blockchain

Our DNS records are kept via a block chain

A. Blockchains in Bitcoin

Bitcoin is a decentralized method of currency. Here we are particularly concerned with bitcoin's mechanism. Bitcoin's mechanism consists of a shared authenticateable transaction record. [9] [1].

Bitcoin's blockchain is essentially a shared ledger. The blockchain is a record of every single transaction made using the Bitcoin system. Each transation refers to previous transactions to indiacate the funds handled by a given transaction are in fact own by the user iniating the transactions. The record is validated by traversing the tree of transactions and marking referenced transactions as used. A valid blockchain has all non-leaf node transaction marked as used only once.

B. Using the Blockchain for DNS records

How does our blockchain differ? Why do we a different blockchain. Ours is smaller because we don't want to keep all those records. That's expensive!

We still mine for blocks. Difficulty still to be determined.

IV. VHASH

VHASH was born of a deficency in current DHT mechanisms to allow for spacial representation of hash locations. The intent is that meaning can be ascribed to locations in the DHT and facilitate more efficent function. Specifically in this example we seek to build a minimal latency overlay network for the DHT so a global scale DHT is viable and efficent. The naive method of doing so is to assign coordiants to servers based on embedding location of nodes on the planet earth. More complex appraches would be to approximate a minium latency space based on inter-node latency.

A. Mechanism

VHASH can be implemented in any arbitrary number of dimensions. The overlay space used by the DHT is comprised of a unit hypercube with toroidaly wapping edges. The toroidal property makes some visualization and mathmatical properties difficult but allows for a space without a sparse edge such that higher dimension graphs/spaces can be embeded with less error.

VHASH nodes are responsible for the address space defined by thier voronoi region. This region is definded by a list of peer nodes maintained by the node. A minium list of peers is maintained such that the node's voronoi region is well definded. The links defined by this set of peers are links on the graphs delunay triangulation.

B. Relation to Voronoi Diagrams and Delunay Triangulation

Vhash does not strictly solve Voronoi diagrams [2]. The toroidal nature of the space precluids from straightforward application of the mechanisms described here[2]. However the algorithim does approximate a graph with similar properties. Rather than attempt to calculate the voronoi region of each node, it simply filters locations assigning responsibility to the nearest node. An online algorithim maintains the set of peers which define the node's voronoi region. The set of peers required to define a node's voronoi region descibe a solution to the dual Delunay Triangulation.

C. Messages

Maintence and joining are handled by a simple periodic mechanism. A notification message consisting of a node's information and active peers is the only maintence message. All messages have a destination hash location which is used to route them to the proper server. This can be the hash location of a particular node or the location of a desired record or service which will be recived by the node responsible for the location. Services run on the DHT define thier own message contents.

D. Message Routing

Messages are routed over the overlay network using a simple algorithim. A node maintains a minimal list of peers to define it's own voronoi region. From the perspective of the node, the entire voronoi region is defined only by its peers.

When routing a message to an arbitrary location, a node calculates who's voronoi region the message's destination is in amoungst the node and its peers. If the destinion falls within its own region then it is responsible and handles the message acordingly. The node otherwise forwards the message to the closest peer to the destination location. This process describes a precomputed and cached A* [5] routing algorithim. A* offers $O(log(n))$ hop routing.

E. Joining and Maintence

Joining is a straightforward process. A new node must know at least one member of the network to join. A joining node choses a location in the hash space as its owner location at random or based on a promblem forumulation (for example based on geographic location) and the joining node sends a maintence message to it's own location via the known node. The message is forwarded to the current owner of that location who can be considered a "parent" node. The parent node imediately replys with it's own maintence message containing it's full peer list. The new nodes final peers are a subset of the parent and the parents peers. The parent adds the new node to it's peer list and removes all occuluded nodes from the peer list. Then regular maintence propogates the new node's information and repairs the overlay topology.

Maintence is a periodic process in which each node sends its peers a maintence message. The maintence message consists of a nodes information and the information of that nodes peers. When a maintence message is recived, the reciving node considers the listed nodes as canidates for its peer list and remove occluded nodes from thier own peer list. When messages to a peer fail it is assumed the peer has left the network. The leaving

peer is removed from the peer list and canidates from the set of 2 hop peers provided by other peers move in to replace it.

There is no protocol for polite exit from the network as it is unessisary. The DHT protocol assumes nodes will fail and the diference between an intended failure and unintended failure is unessisary. The only issue this causes is that software should be designed to fail totaly when issues arrise rather then attempt to fulfil only part of the protocol rather then to partialy fail and result poor coverage of the region for which it is responsible.

F. Worst cases

Our worst case scenarios only occcur in contrived scenarios and are astronomically improbable assuming a random distribution of points.

V. P^{3}DNS

P^{3}DNS has logically descrete components which provide DNS efficent record storage, Domain name ownership management and verification, DNS backwards compatitibility, all of which may be logically replaced or have individual optimizations. P^{3}DNS uses a DHT to store DNS records in a distributed fashion, A blockchain and Namecoin[1] analog to manage domain name ownership.

P^{3}DNS utilizes public and private key encryption for signing and verifying records.

A. Distributed Hash Table

Our implementation is not specific to any particular Distributed Hash Table. We examined using Chord [13] with DNS, similar to DDNS [3]. However, Chord's unidirectional ring overlay topology does not take actual network topology into account and using it for a global scale system is not viable because messages will be routed very inefficiently. A DHT which allows the routing overlay to be optimized to the network topology and conditions in real time is required.

As a result we chose to develop a prototype DHT to meet this requirement to act as backend to our DNS system called Vhash. Vhash is built on the idea of an "overlay space" which is a k-dimensional unit cube which wraps around the edges in a toroidal fashion. Each record in the DHT is assigned a location in that space. Each node is assigned a location and is responsible for records to which it is the closest node. The variable dimensionality is allowed so that problems can be embedded into the space with relative ease and records can be assigned locations which have meaning concerning the problem in which they are a part. This way, records that are close to each other in the problem formulation are close to each other in the DHT and are likely hosted on the same node. This offers speedup for many distributed algorithms which require traversal of data.

B. DNS frontend

Because this system is intended to be reverse compatible with the existing DNS protocol, we serve the data provided by the DHT after it has been authenticated by the block chain to other DNS servers or clients. DNS nodes incorporated into the DDDNS system will not request data from other DNS servers and will only exchange data via the DHT

VI. Usage

Im not sure what to put here, this indicates it needs to be re-organized

A. Establishment of a New domain

Under the current DNS system, a new domain name is purchased from a company registered with the Internet Corporation for Assigned Names and Numbers (ICANN). That company adds the domain name and a record provided by the owner to the TLD servers. The owner or management company then maintains a name server to answer DNS requests for the purchased domain. In P^{3}DNS new domain names are instead awarded as part of the blockchain mining process or purchased from a previous owner, then transferred to the new owner. These assignments and transfers are both recorded in the blockchain.

A prospective domain owner can create their own mining software and mine a domain name or purchase a domain name voucher from a miner and exchange it for a domain name. In the blockchain, domain name owners are referred to by their public key which is used to authenticate records and transfer domains. Loss of the private key of an account will result in the loss of ability to update DNS records and ability to transfer the domain.

B. Updating records for a domain

A domain name record in the current DNS system is used to configure a record on your own Name Server or to configure the record held by the TLD server to contains an address record. Using P2PDDNS all records must be signed using their owner's private key.

A properly configured P2PDNS server should not accept a DNS records which has not been signed by

its blockchain confirmed owner or accept a record with an older version number. To push a new DNS record for a domain the owner must create the record set for the domain and then sign and submit it to a node on the DHT. The DHT will forward the record to the responsible party and store it after confirming its validation. The new record will begin to be broadcast to clients after old records begin to expire and caches seek new records.

C. Looking up a DNS record

In the current DNS a record is looked up using a UDP system that queries a tree of requests. clients send queries to a local DNS server which acts as a resolver and cache. If a portion of the domain name is unknown the resolver sends a request to the responsible server and looks up recursively from there.

This system is largely unchanged under P^{3}DNS from the point of view of the resolver and client. Ideally the resolver or client has chosen a nearby member of the P^{3}DNS network as its root domain server (it can also maintain a large list of backup servers should its current one fail). The resolver requests a domain's record from the P^{3}DNS node. The node then forwards the request to the responsible party. If any node along the route has a valid cache of the required record, then that server responds and routes the message back to the entry node.All nodes along the route cache the response to aid future queries.

D. Caching

DHT(DNS?) needs to aggressively cache lookups. Previous investigations into optimizing caching on a DHT saw good results [11] however with the sepcific application of DNS in mind the time to live field on DNS records deffers the caching optimization problem onto users who we may or may not trust.

Integrated File Replication and Consistancy Maintanence (IRM) [11] views the process of caching and keeping the cache up to date as components of larger problem. Nodes in the DHT keep track of how often records are requested and cache those records once a defined rate is passed. Nodes then request an update for the cache based on how often the record is requested and how often that request is expected to be changed by the owner.

In the current implementation of DNS, (who does it) caching involves a time to live field defined by the domain owner. This means that there's not reall any sort of cache optimization done by the server; rather the server that thew records have a sensible time-to-live value.

IRM [11] can be used to approximate the correct time-to-live value.

E. The Big Picture

Using all of these components together allows us to create a system with the following features:

- Robustness - The DHT and Blockchain are both robust to failures and attacks
- Extensibility - The DNS reverse compatibility allows any DNS extension to be utilized, if dynamic resolution is required a name server record can be stored in the DHT to point to a user's specialized DNS servers.
- Decentralization - Both the DHT and Blockchain can operate without the support of any controlling organization, this offers security against corruption and abuse.

VII. Conclusion and Future Work

References

[1] Namecoin. https://github.com/vinced/namecoin. Accessed: 11/29/2013.

[2] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991.

[3] Russ Cox, Athicha Muthitacharoen, and Robert T Morris. Serving dns using a peer-to-peer lookup service. In *Peer-to-Peer Systems*, pages 155–165. Springer, 2002.

[4] Steve Crocker, David Dagon, Dan Kaminsky, DKH Danny McPherson, and Paul Vixie. Security and other technical concerns raised by the dns filtering requirements in the protect ip bill. *White Paper*, 2011.

[5] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.

[6] Mark Lemley, David Levine, and David Post. Don't break the internet. *Stanford Law Review Online*, 64:34, 2011.

[7] Paul Mockapetris. Rfc 1034: Domain names: concepts and facilities (november 1987). *Status: Standard*, 2003.

[8] Paul Mockapetris. Rfc 1035: Domain names: implementation and specification (november 1987). *http://www.ietf.org/rfc/rfc1035.txt*, 2004.

[9] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1:2012, 2008.

[10] IP PROTECT. Preventing real online threats to economic creativity and theft of intellectual property act of 2011. s. 968. In *112th Congress, 1st Session, May*, volume 26, 2011.

[11] Haiying Shen. Irm: integrated file replication and consistency maintenance in p2p systems. *Parallel and Distributed Systems, IEEE Transactions on*, 21(1):100–113, 2010.

[12] Lamar Smith. Stop online piracy act. *US Government.[Links]*, 2011.

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