# COMP 535

DNS Readings

Simon Foucher (260 223 197)

**McGill University
Monday, Sept 21st 2009**

In the early ARPA NET design, name resolution was done by consulting a simple text file located on a global server: HOSTS.TXT. As the internet’s popularity expanded, so did its physical infrastructures, as well as the number of queries sent out to access HOSTS.TXT. It soon became apparent that a better name resolution scheme was needed, so the DNS system was designed. DNS was meant to be a general purpose database used primarily to translate human-friendly names of internet resources into their corresponding IP address.

The DNS system is composed of 2 entities: Nameservers and Resolvers. The name servers are static distributed servers whose primary job is to contain mapping information between domain names and a Resource Record (RR) which contain valuable information such as IP address associated with the domain name, mail host, owner, etc…The query system to the databases was design to accept a name as well as the type of information required. Each RR also carries a field called ‘Time To Live’ (TTL), which is assigned with a default value of 2 days by the domain’s authoritative name server upon a query. RRs are cashed until their TTL expires, which encourages the use of cashed information to speed up the name query process.

The DNS name space is hierarchically distributed between non overlapping regions called domains. The responsibility for a domain is distributed between one or several authoritative nameservers. The nameservers are responsible to manage the information contained within their domain, and can delegate the responsibility for sub-domains rooted within their domains to other name servers. This creates a hierarchy of dependency between domains, with the highest figure found at the global root domain. The root servers are authoritative for TLD servers who manage top level domains like .com, .net or .ca. The TLD servers delegate the management of regular domains to other name servers, who can in their turn delegate the management of sub-domain space.

 The second important entities in DNS are the resolvers. They send queries to the name servers on behalf of clients by following the chain of authority. The resolver first contacts the root server to resolve the TLD, then contacts an authoritative server in the requested TLD, who forwards the request to a name server, and so on until the original query is matched with the requested RR. Since this system incurs significant delays, aggressive cashing is implemented by the authoritative name server by applying a TTL stamp on the data. Negative response to queries is also cashed, which has been found to reduce the load on nameservers considerably.

 There are many issues with the current DNS infrastructure. One of which comes from the fact that distributing authority across the name space did not distribute a corresponding amount of expertise, and some nameservers administrators have been arbitrarily increasing TTLs to weeks, if not years, in an effort to enhance performance. This resulted in a broken link between upstream changes and downstream users, who will get misdirected until the TTL has expired. Since DNS does not keep records of cashed copies sent out, widespread cashing stalls the propagation of unanticipated changes in the domain infrastructure. At the other end of the spectrum, large scale data distributers like Akamai only cash DNS queries for 30 seconds in order to provide extremely up to date data, but the flood of DNS request congestions the affected nameservers and slows down the whole DNS infrastructure.

 Another important issue with the DNS system is its vulnerability to DoS attacks. Even though all authors agree this point, they have diverging opinions on the source of that vulnerability. One author argues that a domain’s vulnerability is proportional to the quantity of authoritative nameservers serving it. The author mentions that sharing the authority for a domain amongst many different nameservers increases the chances that at least one of those servers has a security flaw which can be exploited. Then, by launching a DoS attack on the non compromised name servers, an attacker could hijack that domain since all the legitimate queries would be forwarded to the compromised name server by regular load balancing algorithms. A lot of those vulnerabilities come from older versions of the BIND software, developed by Bakerley University in the early days of DNS as a Unix solution for nameservers. The second author argues that the more a domain has nameservers serving it, the less it is vulnerable to be taken down by an attacker. If a domain is only served by one or two name servers, an attacker could easily launch a DoS attack on that/those nameservers and effectively take down DNS access to that domain. Both authors present some very similar statistics on the hierarchical distribution of the DNS domain space, but their interpretations are contradictory.

 There seems to be a consensus that the DNS system is far from perfect and could use some enhancements. Its hierarchical nature makes it vulnerable to DoS attack and may lead to slow access in congested periods. One author suggests an alternative to DNS, which is a peer-to-peer based system. The use of a P2P name query network would effectively spread the work load across many servers, which should both enhance performance and make it rather difficult to launch targeted DoS attacks. This ‘CoDoNS’ has been applied as a voluntary alternative to DNS for the moment, and is currently being tested. If it does work correctly, then ‘internet managers’ might consider getting rid of the old DNS and implement this new system permanently.