6.033 2022 — Recitation 2: DNS

DNS is a big system! These notes cover the highlights. You should make sure you're comfortable with the examples covered in Section 4.4 of the textbook. Come to office hours, especially our Sunday office hours, with questions if you have them!

Starting point: Basic DNS lookup. See the Lecture 2 notes for this. Important points: for a lookup of eecs.mit.edu, the root nameserver will refer the client to the edu. nameserver; that nameserver will refer the client to the mit.edu. nameserver; that nameserver will be able to answer the query for eecs.mit.edu.

- Pro: Delegation; no nameserver has to keep track of the entire hostname-to-IP mapping
- Cons Lots of queries in general, lots of queries to the root specifically

Enhancements (not the best term, these are pretty crucial to DNS functionality)

- Initial query can go to any nameserver, not just the root
- *Recursive* queries: a nameserver can issue queries on behalf of a client. E.g., if a client queries a nameserver for the hostname ginger.scholarly.edu, that nameserver can issue the query to the edu. nameserver, and then the subsequent query to the scholarly.edu. Nameserver, rather than making the client perform both.

It's not clear yet why these enhancements help! We still have a lot of queries, still seems like we need to go to the root for everything. So in comes:

- Caching. DNS clients and nameservers keep a cache of known mappings. A mapping expires after a time set by the nameserver in charge of that mapping.

Combining these three things:

- Suppose a client queries the cse.pedantic.edu. nameserver for the hostname ginger.scholarly.edu; assume the nameserver is set to handle recursive queries. That nameserver will walk the DNS tree on the client's behalf, and learn:
    - The nameserver for edu.
    - The nameserver for scholarly.edu.
    - The nameserver for scholarly.edu.
    - The address of ginger.scholarly.edu.

  It will store all of those results in its cache. So if another client comes along shortly afterwards, which a query for www.mit.edu, the nameserver already knows the edu. nameserver.

These enhancements allow DNS to scale to the size of the Internet. A drawback of caching: if a hostname-to-IP address mapping changes, the change won't go fully into effect until all of the cache entries for the previous mapping have expired; and these cache entries could exist in *many* (millions) of DNS clients/nameservers. The people in charge of DNS nameservers know this, and can set short cache-expiry times for mappings that change often, and can also keep multiple servers alive (the old and new one) until all cache entries have expired.

- When we get to the distributed-systems part of 6.033, we'll be able to talk about DNS as providing — or not providing — particular forms of *consistency*.

Discussion

- **Who's impacted by DNS?** Tons of people. You. Internet users, developers, nameserver operators, anyone who uses the internet, ICANN, small businesses, large businesses,…
- **What's good about DNS's design?** It scales thanks to its hierarchical design (and things like caching), it's reliable in many ways (multiple physical nameservers per zone, e.g.), it's fairly high-performing for a system so large, nice delegation
- **What's bad about DNS's design?** Many people consider its lack of security and trustworthiness to be a downside. We'll come back to this later in 6.033, but as described here, DNS doesn't do any checking to make sure that a nameserver returns the correct hostname-to-IP-address mapping, and that can lead to attacks.
    - *DNSSEC* is a system that addresses this, and we'll discuss that in the last part of 6.033.