Reading Report #11
Paper: Development of the Domain Name System
Student: Shuo Yang

DNS was designed to address the limitation of using HOSTS.TXT for name resolution. It is quite clear that DNS is a better choice than HOSTS.TXT for fast-scaling Internet. However, it is not clear, according to the paper, why DNS stands out as a better design compared to other alternatives? The strengths of DNS lie in its use of distributed database, hierarchical name space and caching. But there seems to be also some drawbacks to the design decisions made for DNS.

The Internet was designed to be resilient under failures, but the design of DNS seems to put single points of failure into the Internet, that is, the root servers. Though root servers can be replicated, but what if they all go down? Then we either rely solely on cache or we get stuck. What if they all contain configuration errors such that we cannot get a correct IP address for a specific host? Then even the physical connections are there, we still get stuck. The design of DNS strikes a balance between centralized and distributed management, a more distributed way of managing this may be needed for the Internet to survive under hard failures (e.g, server goes down) or soft failures (e.g, configuration errors).

The original DNS design puts a heavily focus on usability, without mentioning even a little bit about security, which is now a huge problem for DNS. The well known DNS attack targets exactly those DNS vulnerabilities. These attacks take advantage of the communication back and forth between DNS clients and DNS servers. Now we have to put additional efforts in making DNS lookups secure. But if we could put security as a primary design principle, we could end up getting less trouble. The design of DNS assumes mutual trust between clients and servers. This assumption, though holds for old days, now opens the door for various of attacks. We need secure protocols for clients and servers to verify identify and exchange information.

The design of DNS adds a level of indirection, that is, just for name resolution. This brings concerns of performance. Indeed, nowadays, often time it is slow DNS lookup that causes the slow service response rather than other factors. Your requests may go back and forth between many DNS servers, from root server to local servers at different levels, to eventually get the results back. The question is: is this extra level of indirection really needed? Can we perform name resolution at a higher network layer with better security, protocol and performance for similar service? I believe answering this question can lead to better design for Internet name resolution.