Paper Critique: DNS

1. Summary of the System: What is it? What does it do?

DNS stands for the Internet Domain Name System. It is a system that combines both the client server system and name scheme implementation in a successful way. DNS, as our textbook says, is a name management and resolution system that uses hierarchy to distribute the management of names among naming authorities and to distribute jobs of resolving names. DNS provides an Internet addresses to each domain name, although a single domain name may contain several Internet addresses. It is a naming system, in short.

1. Does DNS respond to or improve on a previous technique? How?

DNS actually does improve upon traditional naming schemes. It does use several aspects of the traditional naming schemes such as the hierarchical naming scheme, but one thing different is that if omits the context argument when resolving the value or the internet address. When we use in the DNS\_RESOLVE(domain\_name) to get the value of our name, we only pass in the name instead of both the name and context. The reason is stated in section 4.4 of our book as there is only one single, universal, default context for resolving all Internet domain names. Thus, we would only need to pass in the domain\_name. Another change is that DNS does not use BIND and UNBIND procedures. It uses a text editor or database generator that creates and manages binding tables. By changing this procedure, a downfall was that bindings can take up to hours instead of the usual seconds.

3. What are the Modules of the system and their interactions?

DNS uses the distributed directory service model. Thus, the modules are going to be the different servers that are linked together in DNS. We have the root name server, and then we have a higher level name server that gets more and more specific with our domain name. The way they interact is that, we first find the root server, then we resolve our domain name and find the next higher level name server. This happens recursively until either we reach our final name server for our domain name, or DNS finds that the domain name actually does not exist. One last module in DNS is probably the cache system. The cache helps speed up name requests we got in the past by storing the values in a cache. DNS can look to the cache to find domain names and their IP addresses to speed up look up times.

4. What are the design goals of DNS?

The design goals seem pretty simple from the text. The goals were to create name-resolution system that used hierarchy in storing different servers. They also split into several modules in order that DNS be scalable for very large numbers. A big goal in design for DNS was probably to have a successful implementation of a naming scheme along with keeping an excellent client server application. Other goals of the system were standard system goals of scalability, security, and fault-tolerance. On page 175, we get a clear definition of what the goals were such as performance speed, scalability, and resilience.

Scalability:

DNS was scalable because it used something called the distributed directory service model. What this did was have several servers that were responsible for resolving a set of domain names. Thus, each server served the same function and had its delegated tasks. Due to this design choice, DNS was able to respond to increases in scale while maintaining reliability and performance as our textbook says in page 177. As a result, DNS is able to scale to extremely large numbers.

Security:

Although DNS tried to provide some security, they did not provide authentication to the name responses. As stated in the last page of section 4.4, intruders could hide themselves as DNS servers and send out malicious content and malevolent responses to name resolution requests. Even till now, there is no definite way to fixing this problem. Users just have to treat all responses as potential hints and verify them independently.

Fault-tolerance/ handling failures:

DNS did report to have some failure handlers. If DNS could not match the original name with a name service or domain name, it returned a response of “no such domain.” This error message was shown on page 177 of the textbook. However, we would not know if we simply mistyped the domain name or if the domain actually did not exist. A much better error message could be made to handle such failures such as giving suggestions of domain names the user could have meant.