COS332 – Computer Networks: Chapter 3

Brief overview of the DNS (Domain Name System)

- The Domain Name System (DNS) translates computer names to addresses
- In the past, a file on a central server contained two columns: one with addresses, and the other with names. This file was downloaded regularly from a central server and stored on each computer for local name translations.
- These files are still in use today and can be found as /etc/hosts on most Unix systems and as C:\windows\system32\drivers\etc\hosts on most Microsoft Windows systems.
- A typical entry in the file looks like this: 127.0.0.1 localhost. This means that the name localhost corresponds to the address 127.0.0.1 on the computer where this hosts file is located.

Domain Names

- **Domain Name System (DNS)**:
    - DNS uses a hierarchical name space for various parties to manage their part of the name space, a principle often referred to as *local autonomy*.
    - The name space starts with Top-Level Domains (TLDs) which are classified into generic (gTLDs), country-code (ccTLDs), and sponsored (sTLDs).
    - Control over ccTLDs is delegated to each specific country. For example, .za for South Africa is controlled by ZADNA (.za Domain Name Authority).
- **Second-Level Domains**:
    - Once a national authority delegates authority for a second-level domain to an organization, that organization controls the second-level domain.
    - Registries or registrars are authorities for domain names that are mainly used for the registration of further subdomains.
    - Registries can apply additional policies that are not required by the registries above them.
- **Domain Registration and WHOIS Protocol**:
    - To register a domain, you need to provide the registrar with information about your new domain and possibly pay a registration and/or annual fee.
    - The WHOIS protocol is a useful tool for determining who owns or controls a domain. Most registries run a WHOIS server, and WHOIS clients are available for most operating systems.
    - For operating systems that don’t include a WHOIS client by default, ownership information can be obtained via the web. Starting at IANA to determine who manages the TLD is a good approach.
    - Due to abuse of this information, many domain owners use redaction services to display a somewhat anonymous party who acts as an intermediary between official users of ownership information.

More on the various domains

- **Top-Level Domains (TLDs)**:
    - **gTLDs (Generic Top-Level Domains)**: Managed globally. Examples include .com for business, .gov for the US government, and .net for network infrastructure providers.
    - **ccTLDs (Country-Code Top-Level Domains)**: Managed by each specific country. Example: .za for South Africa is controlled by ZADNA (.za Domain Name Authority).
    - **sTLDs (Sponsored Top-Level Domains)**: Managed by specific communities, professions, or companies. Examples include .pro for professional purposes, .expert for experts, and .google for Google.
- **Second-Level Domains in .za (South Africa)**:
    - **ac.za**: South African universities, research institutions, and related organizations.
    - **co.za**: Open to anybody. Managed by UniForum SA.
    - **edu.za**: Further education and training institutions.
    - **gov.za**: South African government.
    - **law.za**: Attorneys.
    - **mil.za**: South African Department of Defence.
    - **nom.za**: Individuals.
    - **org.za**: South African non-commercial organizations.
    - **school.za**: Schools.
- **Root Authority**:
    - **IANA (Internet Assigned Numbers Authority)**: Operated by ICANN (Internet Corporation for Assigned Names and Numbers). Maintains a list of the TLD registries.

How the DNS works

- **DNS Description**: DNS can be described by explaining how information should be prepared to put it in the DNS, or by describing how information may be retrieved from the DNS.
- **Iterative Approach**: The process involves describing how information is prepared for the DNS, followed by an explanation of how such information could be retrieved.
- **DNS Exploration**: Readers are encouraged to explore the DNS by issuing queries like those presented in the examples on systems that they have access to.
- **DNS Structure**: The DNS is implemented as a tree using pointers to link parent nodes to child nodes. The data structure used to represent information at each node is known as a *zone file*.
- **Zone File**: The core of each zone file is a set of triples. Each triple consists of a name, a record type indicator, and an address. The name is the name of a host or of a subdomain. The record type indicator is either A (for address) or NS (for name server). If the record type indicator is an A, then the name that this triple defines is the name of a computer, and the

third field of the triple is the address that corresponds to the name. If the record type indicator is NS, then the name that is being defined is a subdomain and the third field is the address of the server that knows about the names in this subdomain.

- **Example**: For instance, consider the zone file of xx.co.za. It may have the following entries:
    - arthur A 10.1.1.1
    - guinevere A 10.1.1.2
    - lancelot A 10.1.1.3
    - table NS 10.1.1.4

These entries indicate that the hosts 'arthur', 'guinevere', and 'lancelot' have the respective IP addresses 10.1.1.1, 10.1.1.2, and 10.1.1.3. The 'table' entry is a subdomain with the name server located at IP address 10.1.1.4.

Name resolution in the DNS

- **Name Resolution**: The process of determining what address corresponds to a given name is known as *name resolution*. For example, if a server is asked to resolve the name arthur.xx.co.za, it will respond that the address is 10.1.1.1.
- **Applications and DNS Requests**: Most applications, such as web browsers and email software, must resolve the names they encounter. Therefore, in addition to sending HTTP requests, a web browser will also send DNS requests.
- **lookup Application**: The application lookup is available on most systems for querying the DNS. Linux users may prefer to use dig.
- **Recursive and Iterative Name Resolution**: Name resolution may be performed in a recursive or iterative manner. In the recursive case, a name server is asked to resolve a name from the root to the lowest level of the name. For iterative name resolution, a name server will refer one to other name servers as required, and one therefore must use that information to traverse the tree iteratively.
- **Root Name Servers**: There are 13 root name servers on the Internet, known as a.root-servers.net through m.root-servers.net. These root servers only know where the name servers for the TLDs are. They won't provide any other information requested from them.

Resolving domain names iteratively

- **Using nslookup**: You can use nslookup to change the default name server and set the type of DNS record you want to query. For example, you can use server 192.5.5.241 to change the default server to f.root-servers.net and set type=ns to specify that you want a pointer to a name server.
- **Fully Qualified Domain Names (FQDNs)**: A FQDN specifies the domain name starting at the TLD down to the domain name of interest. A full stop at the end of a FQDN unambiguously indicates that it is a FQDN.

- **Redundancy**: The existence of multiple name servers for a domain and the 13 root name servers provides redundancy. This ensures that if one server fails, a client can simply fail over to the next server.
- **Example of Iterative Name Resolution**: The following is an example of how to use nslookup to iteratively resolve the name www.example.com. This involves querying the root server for the .com TLD, then choosing one of the name servers for the .com domain (e.g., A.GTLD-SERVERS.NET), and finally querying this server for example.com.

Zone Files

- **Zone File Serial Number**: The serial number of a zone file is used to track versions of the file. A newer version of the zone file always has a number that is larger than an old version. The serial number is often formed using the date of the change in the format YYYYMMDDnn, where YYYYMMDD is the date and nn indicates that it is the nnth version of the file for that day.
- **Primary and Secondary Servers**: The primary server, also known as the master server, is the one that loads the zone file from disk. Secondary servers, also known as slave servers, contact the primary server to check whether their copy of the zone file is up to date. The copying of zone information from a primary to a secondary server is known as a *zone transfer*.
- **Refresh and Retry Numbers**: The refresh and retry numbers in the SOA record deal with the frequency of zone transfers. The secondary server will contact the primary server every refresh number of seconds to get the latest version of the zone. If the serial number is unchanged, it is not necessary to transfer the entire zone. The retry number indicates the time (in seconds) that a secondary server should wait before trying to transfer the zone if an earlier attempt fails.
- **Expire Value**: The expire value indicates how long secondary servers should consider their data useful. If the expire time is reached, the secondary server will discard its copy of the zone file and will stop answering queries about the names in the zone.
- **Minimum TTL (Time to Live)**: The minimum TTL number indicates how long a client (rather than a secondary name server) may use data it got from a name server before it should get new data. Each of the other records in the zone file — often referred to as resource records (RRs) — may state its own TTL. The minimum TTL will therefore only affect those RRs that do not specify a TTL, or that specify a TTL that is shorter than the minimum TTL.

Roles of Primary and Secondary servers

- **Non-Authoritative Responses**: Some responses are marked as non-authoritative by nslookup if they are answered from cache, because the authoritative server may have been updated since the information was cached. However, for day-to-day operation such information is typically fresh enough to be useful because of sensible TTL settings.

- **Email Handling and MX Records:**
    - Modern systems often divert all email destined for an organization to a specific mail server. This server can scan the email for viruses, filter out spam, and then forward the clean email to the appropriate host within the organization.
    - The DNS system provides for failover services. The number next to MX in the MX records indicates the priority of the corresponding mail server, with lower numbers taking precedence.
- **CNAME Records and @ Entries:**
    - A CNAME (Canonical Name) record assigns an alias to a domain name.
    - The @ symbol caters for an 'empty' entry. This is useful when you want someone to be able to go to the website at xx.co.za (rather than www.xx.co.za).
- **IPv6 and AAAA Records:**
    - An IPv6 address can be specified in the zone file using an AAAA record. In principle, AAAA records work identically to A records, except that they store a different type of address.
- **PTR Records and Reverse DNS Queries:**
    - PTR (pointer) records are used to facilitate reverse DNS queries, where one has the address and wants to determine which name corresponds to this address.
- **DNS Resolution and Zone Files:**
    - DNS resolution follows a path from the root name server to the authoritative name server for the domain in question.
    - The zone file includes an SOA (Start of Authority) record, which provides important information about the domain.
    - The SOA record includes a serial number, which is used to track changes to the zone file.
    - The SOA record also contains refresh, retry, and expire values, which control how often secondary servers check for updates from the primary server and how long they consider their data to be valid.