COMP3331 Week 2 Lecture 2

Application Layer

# DNS: domain name system

DNS is the mapping between IP address and name, it is essentially the mapping service.

# DNS services and structure:

Services DNS provides

* Hostname to IP address translation
* Indirection: replace web server and host the web server somewhere else. Host name can remain the same, but the IP address can be changed to anything (very useful).
* Host aliasing: canonical name, alias name, for example, www.google.com
* Mail server aliasing, a web server mapping to a mail server
* Load distribution: each replicated web server will have its own IP address, DNS will get the closest, least-loaded server

**why not centralize DNS**

* Single point of failure
* Traffic volume
* Distant centralised database
* Maintenance
* Doesn't scale

Three intertwined hierarchies

* **Hierarchical namespace – as opposed to original flat namespace**

Top of hierarchy: root servers

Diagram

Description automatically generated

The root server and top level domain server do not have the mapping information.

* **Hierarchically administered – as opposed to centralized**

Next level: top level domain server: TLD, .com .edu etc

Diagram

Description automatically generated

* **(Distributed) hierarchy of servers – as opposed to centralized storage**

Bottom level: Authoritative DNS servers, provides mapping for name hosts

Diagram

Description automatically generated

Authoritative NS are the ones who manage the actual mapping.

## local name server

* Does not strictly belong to hierarchy, it is like cache and checked if it is stored in the local name server.
* Each ISP has one
* Hosts configured with local DNS server address or learn server via a host configuration protocol (DHCP)
* When host makes DNS query, it is sent to its local DNS server

# Iterated query vs recursive query: DNS name resolution

## Iterated query:

Assume local DNS server doesn’t have the mapping. Then we start from the root, and root server (3) comes back with the TLD DNS server (4), (5) contains the authoritative DNS server name and IP address. Then the local DNS server will talk to the authoritative DNS server

Recursive query:

The requesting host asks question to the local DNS server, assume local DNS server doesn’t have mapping, then it asks the root DNS server, root DNS server asks the TLD DNS server, and then the query is sent to authoritative DNS server. The answer goes back to the requesting host along the same path.

Diagram

Description automatically generated

Figure 1 Iterated Query vs Recursive Query

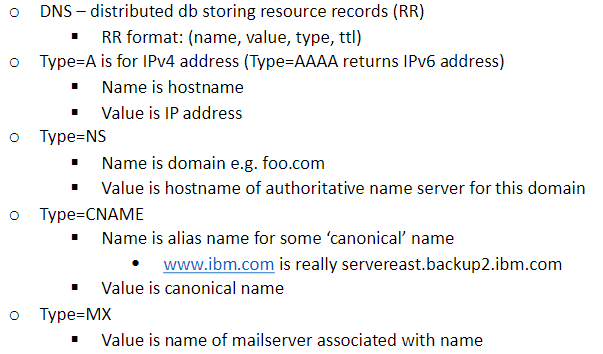
## DNS caching, updating records

Once any name server learns mapping, it caches mapping. Subsequent requests need not burden DNS. Cached entries may be out-of-date, known as TTLs (time to live)

## Type of DNS records

Table

Description automatically generated with low confidence



**Example (recording 1:13-1:17 very important):**

A picture containing text

Description automatically generated

1. The answer sections contain 4 replicate servers, this refers to what we talked about earlier as the load balancing. 4 name servers are sent back. We may get different answers every time we run the query.
2. The authority section is telling the authoritative servers, their names and address

## Inserting DNS records

Register name networkutopia.com at DNS registrar

1. Provide names, IP address of authoritative NS (primary + secondary)
2. Registrar inserts two RR into .com TLD server:

* (networkutopia.com, dns1.networutopia.com (auth NS), NS)
* (dns1.networkutopia.com, 212.212.212.1, A)

1. Any sub-domain, you have to insert record into your auth NS

* Create authoritative server type A record for www.networkutopia.com, type MX record for networkutopia.com

## Updating DNS records

1. Record the current TTL value of the record
2. Lower the TTL of the record to a lower value
3. Wait the length of the previous TTL
4. Update the record
5. Wait for some time
6. Change the TTL back

**Why is DNS using UDP?**

Because it is simple and lightweight. If using TCP, the local server needs to establish connection every time.

# CDN recap

Diagram, schematic

Description automatically generated

## WWW vs non-WWW domains (reading contents, not example)

We can use both, however, www has two advantages:

* Non-www can’t use cname record
* Offloading with www is easier

## Reverse DNS – IP address -> domain name

Special PTR record type to store reverse DNS entries

Where is reverse DNS used?

* Troubleshooting tools e.g., traceroute + ping
* ‘Received’ trace header field in MTP email
* SMTP servers for validating IP address of originating servers
* Internet forums tracking users
* System logging or monitoring tools
* Used in load balancing servers/content distribution to determine location