

Domain Name Service (DNS)

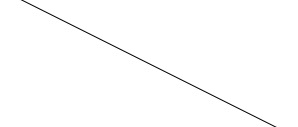
Manas Ranjan Lenka
School of Computer Engineering,
KIIT University

Origins

- People have many identifiers: Full name, Pet name, Roll number, Passport number
- Internet hosts are no less
 - Hostnames and IP addresses

E.g. www.facebook.com

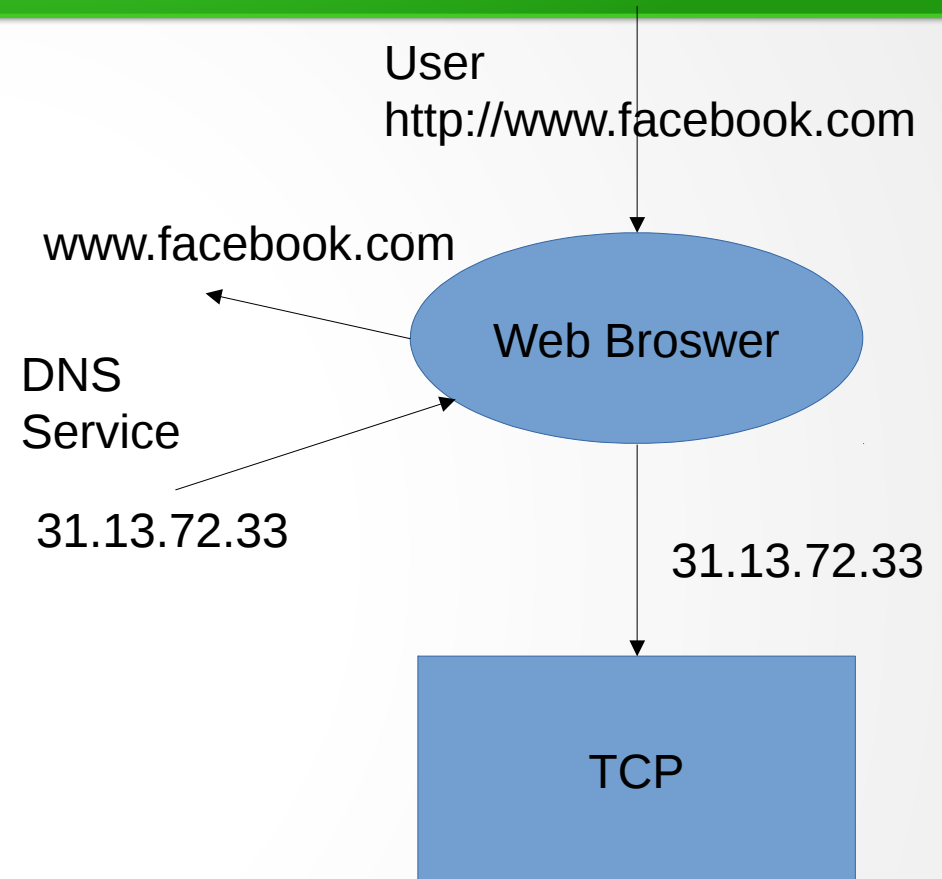
- Variable Length
- Mnemonic
- Carry no info to help route towards them

E.g. 31.13.72.33

- Fixed Length
- Numeric
- Routing packets information embed within them

Problem and Solution

- People prefer hostnames
- Routers prefer IP addresses
- Need a service (DNS) that converts hostnames/domains to IPAddress



Domain Name: Label that defines a realm of administrative autonomy
E.g. facebook.com; iitb.ac.in; mit.edu

DNS Services

- Host name to IP address translation
- Host aliasing: “alias → other names”; many names may map finally to same IP address
 - `www.facebook.com.` (alias hostname) maps to `star.c10r.facebook.com.` (canonical hostname)
 - `www.facbook.com`, `www.facebok.com` map to `www.facebook.com`
 - Helps run multiple services from same server

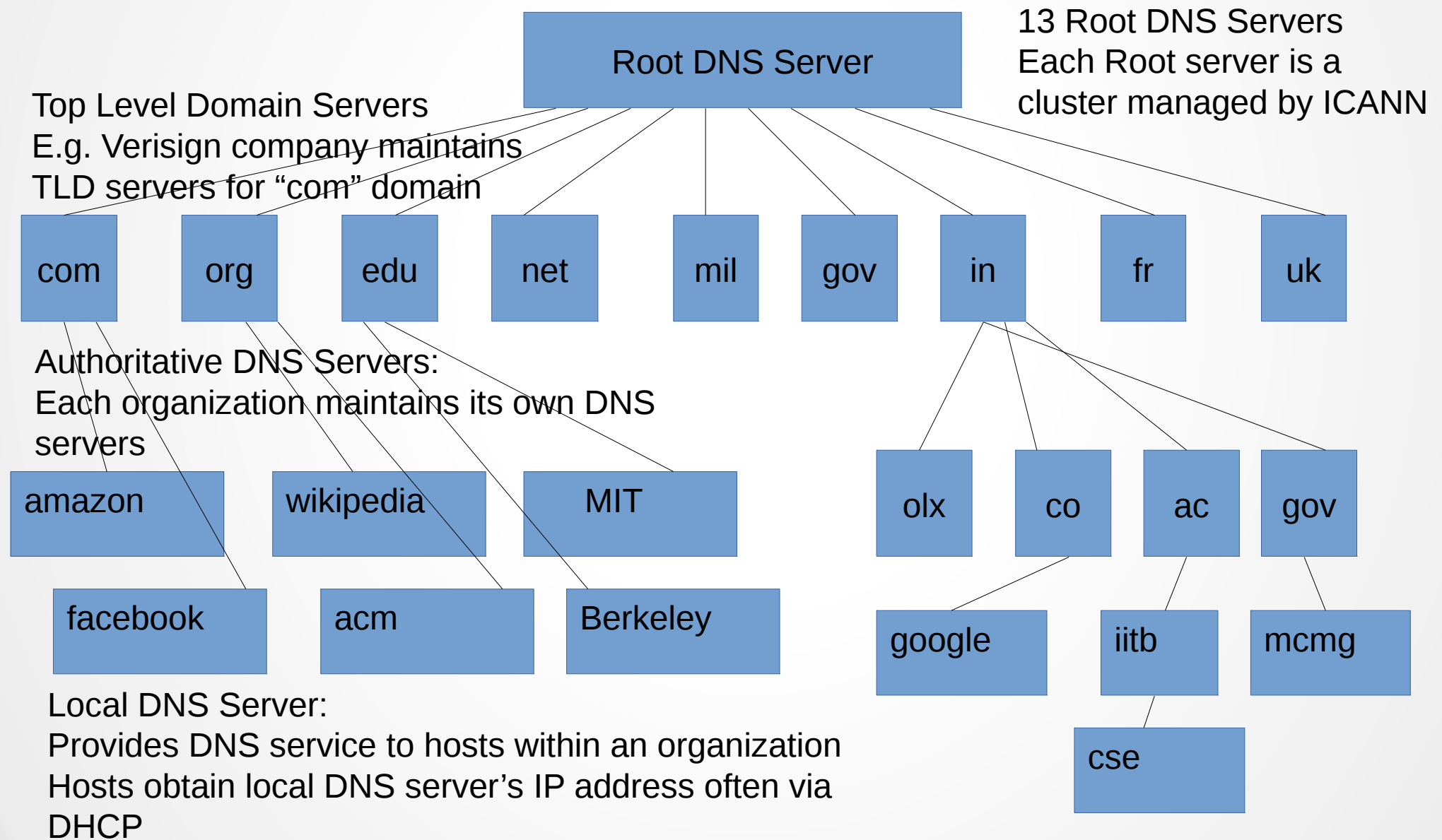
DNS Services Cont...

- Mail server aliasing: Help specify mailserver of a given domain
 - facebook.com maps to msgin.t.facebook.com
- Load distribution: Helps distribute load across replicated servers
 - A single hostname associated with many IP addresses; order rotated on each request

Implementation

- Centralized Architecture:
 - Single point of failure
 - Has to cope with high traffic volume
 - Location: where should it be placed?
 - Huge database maintenance
 - Overall its not scalable

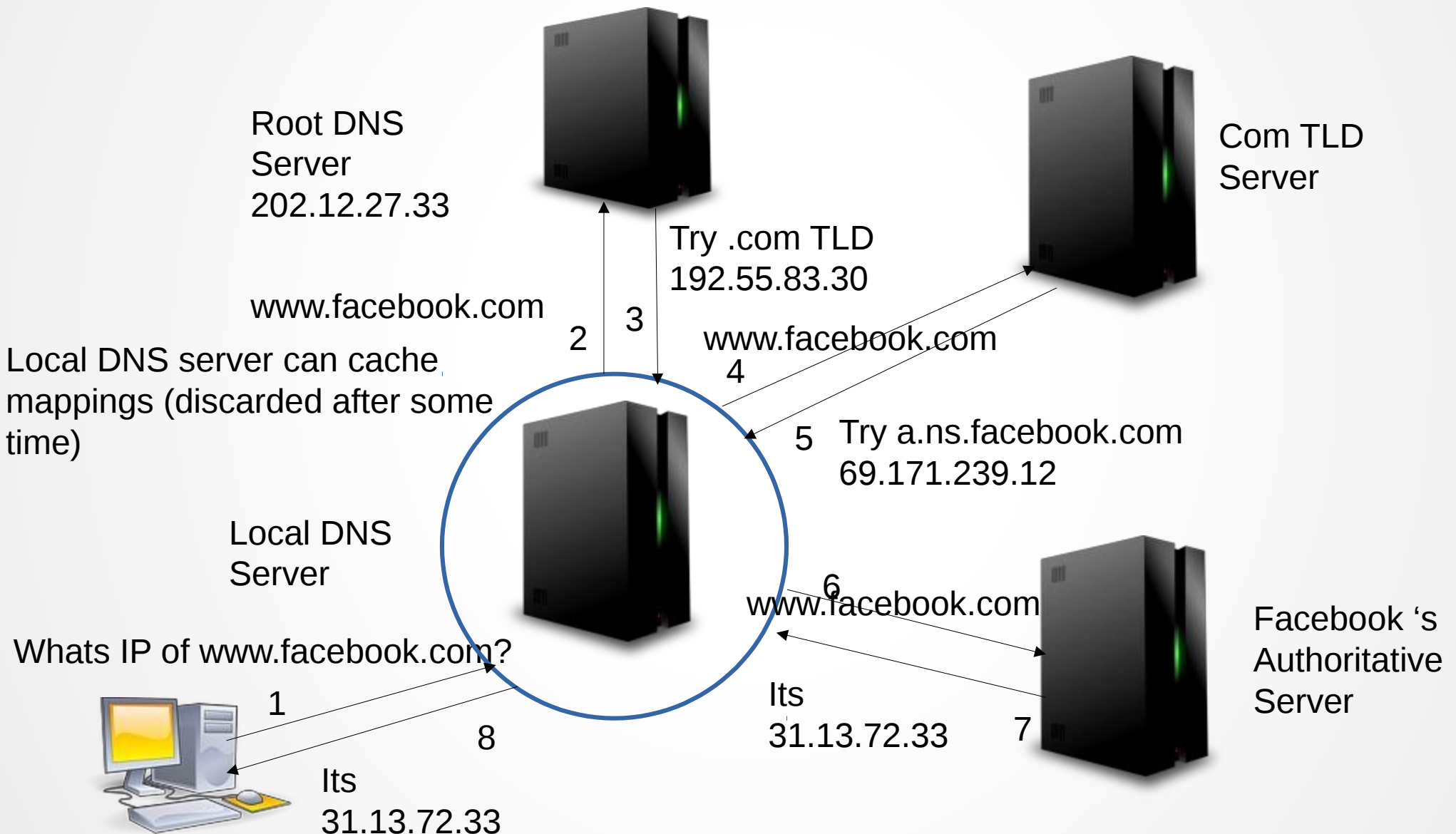
Hierarchical and Distributed Implementation



Root Servers



Example : Hierarchical and Distributed Implementation



DNS Server Database

- Store Resource Records (RRs)
- Four Tuple: [Name, Value, Type, TTL]
- Type=A; Name: Hostname; Value: IP Address
 - E.g. [star.c10r.facebook.com, 31.13.72.33, A, 17]
- Type=NS; Name: Domain; Value: host-name of the authoritative name server
 - E.g. [facebook.com, a.ns.facebook.com, NS, 172797]

DNS Database

- Type=CNAME; Name: Hostname; Value:
Canonical hostname
 - E.g. [www.facebook.com,
star.c10r.facebook.com, CNAME, 2362]
- Type=MX; Name: domainname; Value:
Canonical name of the mail server
 - E.g. [facebook.com, msgin.t.facebook.com,
MX, 300]

Rules

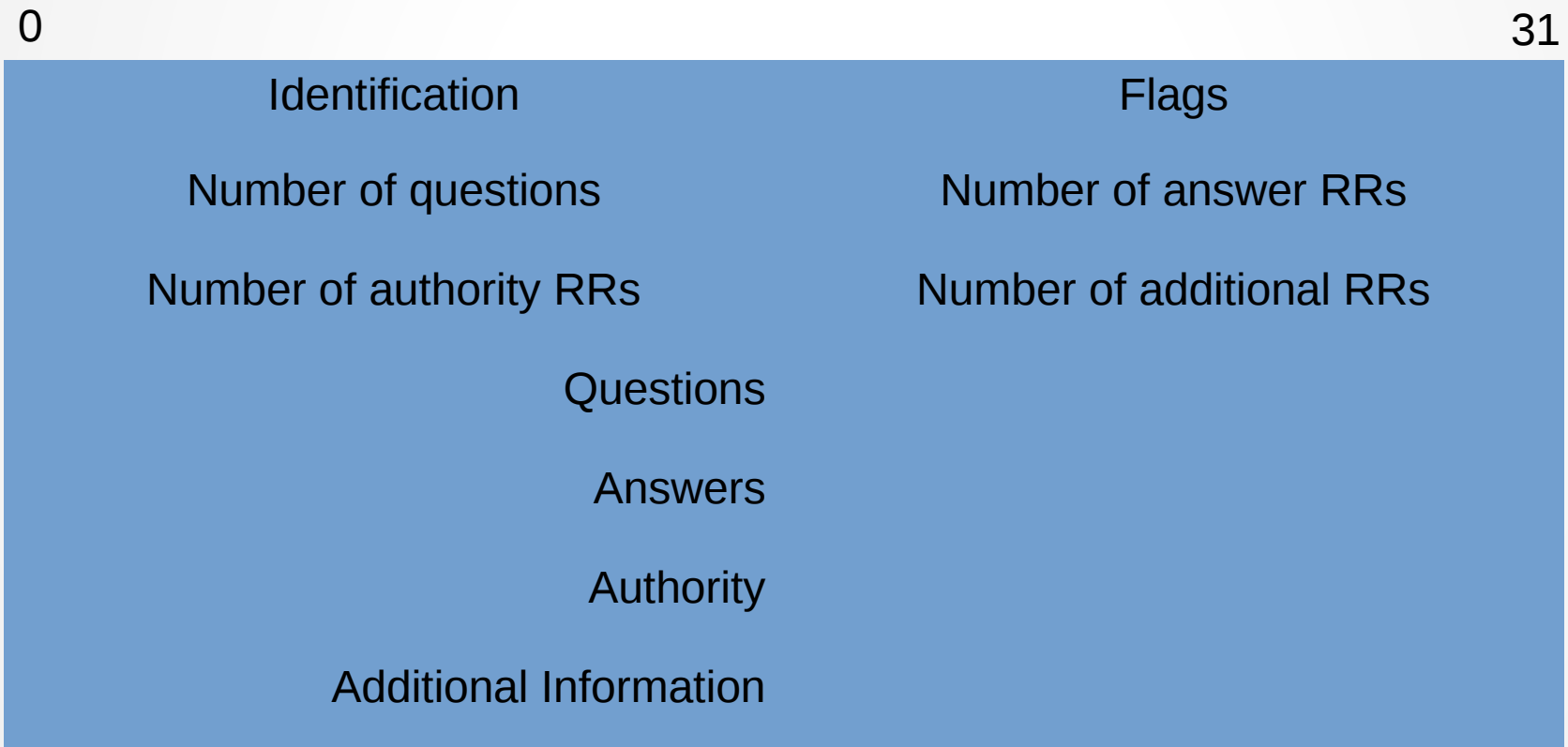
- An authoritative name server (for a given host) will always contain type A record of that host
- A non-authoritative name server will contain a type NS record for the domain and the type A record of the domain's authoritative server
 - [facebook.com, a.ns.facebook.com, NS, 172797]
 - [a.ns.facebook.com, 69.171.239.12, A, 172575]

Inserting Records

- Register domain name with a Registrar
- Provide names and IP addresses of your primary and secondary authoritative DNS servers to Registrar
 - 4 entries (type NS and A) entered in the TLD (or higher up) server
- Within your authoritative server, provide type A and type MX entries for you web and email server

DNS Message Format

- DNS runs over UDP and uses port 53



dig Command Usage

When you pass a domain name to the dig command, by default it displays the A record (the ip-address of the site that is queried) as shown below.

```
$ dig www.gmail.com
```

```
; <<>> DiG 9.9.5-3ubuntu0.1-Ubuntu <<>> www.gmail.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50944
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.gmail.com.                IN      A

;; ANSWER SECTION:
www.gmail.com.                21141   IN      CNAME   mail.google.com.
mail.google.com.             21297   IN      CNAME   googlemail.l.google.com.
googlemail.l.google.com.     4       IN      A       74.125.236.53
googlemail.l.google.com.     4       IN      A       74.125.236.54

;; Query time: 2 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Thu Dec 18 16:17:11 IST 2014
;; MSG SIZE  rcvd: 116
```