Application Layer (DNS, P2P, Video Streaming and CDN)

DNS

Humans can be identified by many ways, our names, our social security numbers, driver's license number etc. Just like how humans can be identified in many ways, so can Internet Hosts. (Hostname e.g. = www.yahoo.com).

- Hostnames provide little information about the location of the host.
- Hosts are also identified by so-called IP addresses, as routers do not understand the hostname string.
- A DNS (domain name system), is essentially a directory service which translates host names into IP addresses.
- It is implemented in a hierarchy of many name servers
- Application-Layer protocol: Hosts, name servers communicate to resolve names
- As end users we don't directly interact with DNS, but all the applications that we use, do.
- Procedure:
    - Client wants to send a HTTP request message to a web server
    - User's host must first obtain the IP address of that website
    - The same user machine runs the client side of the DNS application
    - The browser extracts the hostname, from the url and passes the hostname to the client side of the DNS application
    - The DNS client sends a query containing the hostname to a DNS server
    - The DNS client eventually receives a reply, which includes the IP address for the hostname
    - Once the browser receives the IP address from the DNS, it can initiate a TCP connection to the HTTP server process located at port 80 and that IP address.
- No centralised DNS:
    - Single point of failure
    - Traffic volume (so many queries)
    - Where do we place a single entity in the globe?
    - maintenance
- goals:
    - No naming conflicts, can't have same hostnames
    - distributed, autonomous administration.
        - ability to update my own domain names
        - Don't want to track everybody's updates.
    - highly available

Hierarchy

```
- Hierarchical namespace
    - top level domains are at the top
    - Domains are subtrees
    - Name is leaf to root path
```

```
    - instr.eecs.berkeley.edu -> this is a tree. first we go to the leaf instr
then to the node eecs.... till we reach the root at edu
- Hierarchically administered
    - A zone corresponds to a distinct contiguous portion of the DNS name space
that is managed by a particular administrative authority. Like each of the
namespaces could be within different zones, and each of those zones are managed by
different people.
- Distributed hierarchy of servers
    - Top of the hierarchy: Root servers. The location is hardwired into other
servers.
    - Next Level: Top level domains (TLD) servers
        - .com, .edu,  and these are managed professionally
    - Bottom Level: Authoritative DNS
        - Store the name to address mapping
        - Maintained by the corresponding administrative authority (Service
provider or an organisation)
        - These store the resource records for all DNS names in the domain that it
has authority for.
- Each server can discover the servers that are responsible for the portions of
the hierarchy
    - Everyone knows who the root is
    - Root servers know about all the Top level Domains.
- There used to be 13 root physical servers. They were then scaled because they
worried about them going down. These physical servers are replicated all over the
world. Here in Australia we have 25 root servers. The servers in Australia are
numbered A - N. They all have the same IP address. For example all 'A' servers
have the same IP address, all 'B' servers have the same IP address. Here
anycasting is exploited, whereby if anyone has server 'A' as their destination,
the routers along the way will direct the request to the closest 'A' server.
```

- Host Aliasing:
    - A host with a complicated hostname can have one or more alias names. DNS can be invoked by an application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.

- Mail Server Aliasing: