Domain Name System (DNS)

Motivation

- IP addresses are hard to remember
- Meaningful names easier to use
- Name resolution – map names to IP addresses
- Namespace
  - Flat
  - Hierarchical

Flat Namespace

- Each host given a name
- Special file to keep name-address mapping (ex. /etc/hosts file in Linux)
- All hosts must know the current mapping for all other hosts with which they want to communicate
- Central authority to maintain authoritative host file with which all other hosts sync (HOSTS.TXT at NIC)
- Makes the hostname file too large and the entire scheme unmanageable to be practical in any large network (ex., Internet)

Hierarchical Namespace

- Break complete namespace into “domains”
- Domains broken up recursively into subdomains to create any level of hierarchy
- Delegate task of name allocation/resolution to distributed name servers

DNS

- Naming system for the internet
- Hierarchical naming schemes
- Specifies name resolution mechanism
- Can handle multiple object types within one system
  - "Type" associated with each name to distinguish different types of entities
  - Ex. the name "cse.iitkgp.ac.in" can be a domain name, a simple host name, an email server name etc.
- Large number of RFCs
  - See https://www.statdns.com/rfc/ for a list of DNS related RFCs
  - RFC 1034/1035 has the basics

DNS Names

- Complete namespace is a tree of domains
- Root is a special domain (no name)
- Top level domains – domains at second level of tree
  - *com, edu, gov, net, mil, int, org, arpa, in*, country specific domains (*us, in, kr* etc.)
  - ICANN has authority over all TLDs
  - Delegates management to various entities,
- Domains from third level
  - Managed by local authorities

DNS Names (cont.)

- Every node in the tree has a label (max 63 bytes, case insensitive)
- Sibling nodes must have different labels
- DNS name of a node = sequence of labels from that node to root, separated by '.'
- Absolute names – names that end with '.'
- Relative names – names that does not end with '.', meaning they will be completed by appending something
- Nodes can be domains or hosts
- Arbitrary hierarchy allowed (but implementations usually limit name length to 255 bytes)

- Domain : subtree of the DNS namespace tree
- Zone : subtree for which the naming authority has been delegated to some server
- Domain *x.y* and Zone *x.y* may not be same, as part of *x.y* domain may have its own naming authority and is not part of *x.y* zone

Name Servers

- Contains mapping information for one or more zones (text files in standard format – zone files)
- Maps names to IPs (forward lookup, mandatory) or IPs to names (reverse lookups, optional)
- Primary/Master name server : Name server containing the primary copy of the zone file
  - Can be read and modified
- Secondary name server: pulls zone file data from primary name server (*zone transfer*)
  - Read-only copy
- Authoritative server for a zone: either a primary or secondary server for that zone
- A host can be primary for some zones and secondary for others at the same time

- Note that this primary and secondary DNS server definition is different from the primary and secondary DNS server that is there in the network configuration on your PC/Laptop
  - That just says which server the resolver will contact first
  - Secondary put for fault-tolerance in case primary is down

Root Servers

- Name servers for root zone
- Contains name server for all top level domains
- Currently 13 root servers (*a.root-servers.net* through *m.root-servers.net*) with well-known IPs that can be queried
- All DNS name servers knows at least one root server IP
- But are there only 13? Seems too low!
- No, each root server IP is actually a large number of servers in the background
- See https://www.iana.org/domains/root/servers for more details of root servers

Name Resolution

- Resolver
  - DNS client side program
  - Accesses name server for name resolution
  - Knows the address of at least one name server
  - Sends a DNS request to the name server
  - Standard access routine: gethostbyname()
- Name server
  - Gets request from resolver
  - Looks up the name and sends back response

Name Resolution Basics

- Each domain’s name server must contain the name servers of any subdomains
  - Root servers will contain name servers of top level domains
  - Top level domains will contain name servers for subdomains
    - Ex: name server for .com will contain the IP address of name server of google.com
    - Ex: name server for ac.in will contain the IP address of name server of iitkgp.ac.in
- Name resolution
  - Contact root server for name server of top level domain
  - Name server for top level domain gives name server for next level domain
  - Process continues until mapping is found or error

Example

- To resolve www.yahoo.com, first contact root server to get name server for *com*
- Querying name server for *com* gives IP address of name server for *yahoo.com*
- Querying name server for *yahoo.com* gives IP address of *www.yahoo.com*
- Three queries needed to resolve the name in the worst case

Recursive/Iterative Query

- Recursive Query
  - DNS server either gives the mapping, or forwards the request to the name server that may have it
  - Original requestor finally gets either the mapping or an error
- Iterative
  - If DNS server does not have mapping, it gives the address of the name server that may have it (*referral*)
  - Original requestor contacts the new name server
  - Repeated until mapping is found or no referral is obtained (error)
- Servers must implement iterative query, may implement recursive query (most do)
- What are the pros and cons? Which one should be used?

Caching

- Starting with root server always increases resolution time, increases load on root servers also
- Caching employed at both client and server for efficiency
  - Lookup results in cache (both final IP address, or name server addresses for intermediate domains, for ex. name server for *.com* domain)
  - Answer from cache if found (*non-authoritative* if not authoritative for that zone)
  - Refreshed at regular intervals
- Caching Name Servers: not authoritative for any zone, only caches entries for other zones
  - No zone file of its own

What is in a Zone File?

- Each zone file contains a set of resource records (RRs) for that zone
- Different types of RR's to represent/map different types of things

Resource Records (RR)

- Each RR has: name, type, TTL, Rdata, plus some other fields
- RR Types (16 bit value):
  - SOA : Start of authority
  - NS : authoritative name server for the domain
  - A, AAAA: hostname
  - MX : mail server
  - CNAME : alias name
  - HINFO : CPU and OS Info
  - PTR : pointer to another part of namespace
  - SRV : Service name (RFC 2782)
  - Others….

- TTL : indicates how long the RR can be cached (32 bit integer in seconds)
- RDATA : a type specific value (for ex., an IPv4 address for A type etc.)
- Some other fields in RR not of interest to us

An Example

```
$TTL 3D
@       IN      SOA     mc1.land-5.com. root.land-5.com. (
                        199609206       ; serial, todays date + todays serial #
                        8H   ; zone file refresh period for secondaries
                        2H   ; retry period for secondaries if primary is unreachable
                        4W   ; expiry time if zone file cannot be refreshed
                        1D )  ; minimum TTL of any RR
                NS      mc1.land-5.com.
                NS      ns2.psi.net.
                MX      10  mailsrv.land-5.com.  ; Primary Mail Exchanger
                  MX      20  backupmail.land5.com.
                TXT     "LAND-5 Corporation"
router                  A           206.6.177.1
mc1.land-5.com.         A           206.6.177.2
mc2.land-5.com.         A           206.6.177.3
mailsrv                  A          206.6.177.4
ftp                 CNAME       mc1.land-5.com.
news                CNAME       mc1.land-5.com.
funn                     A          206.6.177.2
```

```
www      CNAME  mc1.land-5.com.
         CNAME  mc2.land-5.com.

telnet.tcp    SRV  10  1  23 mc2.land-5.com.
              SRV   10  3 23  mc1.land-5.com.

subdomain1.land-5.com.  NS   ns1.subdomain1.land-5.com.
subdomain2.land-5.com.  NS   ns2.subdomain2.land-5.com.

ns1.subdomain1     A    202.122.132.7
ns2.subdomain2     A    202.122.136.9
```

Forwarders

- A DNS server X to which DNS queries can be sent by another DNS server Y if it cannot resolve it
- X resolves it and sends back the result to Y. X also caches.
- Motivation:
  - No internet connection for Y
  - Forwarder cache builds up over time
  - Forwarder may be able to resolve most queries
- X may or may not be authoritative for any zone
- Y does not need to know root servers

Protocol Details

- Usually runs on UDP port 53
- Uses TCP for zone transfers (and some large responses)
    - Why not UDP?
- Same message format for query and response

Reverse Lookup

- IP to name mapping
- Not mandatory to implement, but most DNS servers support
- All IP addresses are part of the special zone in-addr.arpa
  - Ex. 10.5.17.2 will map to the name 2.17.5.10.in-addr.arpa
  - PTR type RR kept to map this to a name
  - Lookup similar otherwise

Dynamic Update

- Simple DNS requires the primary name server to be updated manually when a mapping changes – not good for working with protocols like DHCP
- Dynamic DNS updates allow dynamic updates to zone files by messages
- For more details, see RFC 2136