Domain Name System (DNS)

CS60002: Distributed Systems

Antonio Bruto da Costa
Ph.D. Student, Formal Methods Lab,
Dept. of Computer Sc. & Engg.,
Indian Institute of Technology Kharagpur

Motivation

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

Flat Namespaces

- Each host is given a name
- Special file to keep name-address mapping (Ex: /etc/hosts file in Linux does this)
- 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 Namespaces and DNS

- 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 scheme
- Specifies name resolution mechanism
- Can handle multiple object types within one system.
  - "Type" associated with each name to distinguish different types of entitieis.
  - Ex: The name "cse.iitkgp.ac.in" can be a domain name, a simple host name, an email server name etc.

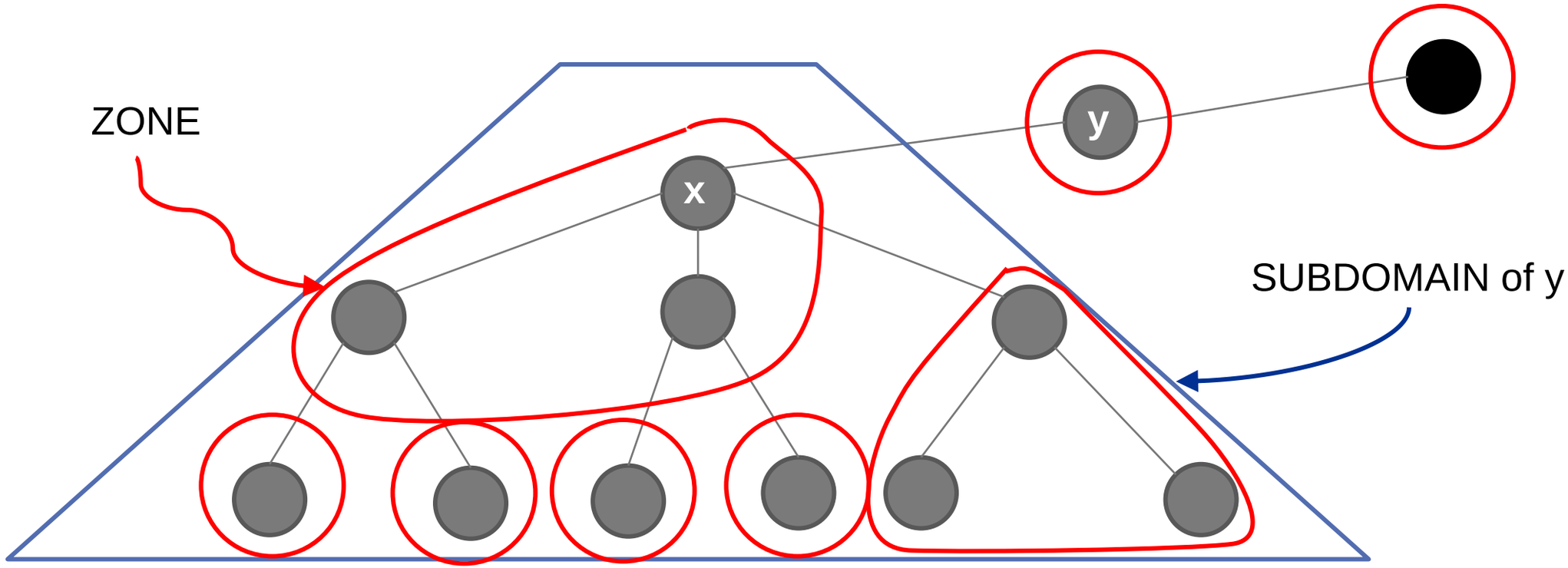
DNS Names

- Complete namespace is a tree of domains
- The Root is a special domain (NO NAME)
- Top level domains – domains at second level of the tree
  - com, edu, gov, net, mil, int, org, arpa, and *country specific domains* (in, us, kr etc.)
  - Managed by NIC
- Domains from third level
  - Managed by local authorities

DNS Names

- Every node in the tree has a label (max 63 bytes, case insensitive)
- Siibiling nodes must have different labels
- DNS name of a node = sequence of labels from that node to the root, separated by '.'
- Absolute name : Names that end with '.'
- Relative names : Names that do not end with '.', meaning they will be completed by appending something.
- Nodes can be domains or hosts
- Arbitraty hierarchy allowed (but implementations usually limit name length upto 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 the same.
  - The x.y domain may have its own naming authority and is not part of the x.y zone.



Name Servers

- Contain 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 lookup, optional)
- Primary/Master name server : gets mapping data for zone from zone file on the host it runs on.
- Secondary name server: pulls zone file data from primary name server *(zone transfer)*
- Authoritative server for a zone: either primary or secondary server for that zone.
- A host can be a primary NS for some zones and secondary for others at the same time.

Root Servers

- Name servers for root zone.
- Contains name server for all top level domains
- Currently 13 “root servers” spread all over the world. (all secondaries of a hidden primary, a.root-servers.net through m.root-servers.net) with known IP
  - Each is actually a named authority, with multiple actual physical servers servicing queries to it.
  - So actually hundreds of physical root servers, but we say 13, as 13 named authority faces with 13 well known IPs.
- All DNS name servers know at least one root server.

Name Resolution

- Resolver
  - 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

- 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 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 Queries

- Recursive:
  - 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.
  - This repeats until a mapping is found or no referral is obtained (an error)
- Servers must implement iterative query, may implement recursive query too (most do).

Caching

- 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 example name server for .com domain)
  - Answer from cache if found (non-authoritative if not authoritative for that zone).
  - Refreshed at regular intervals.

- Caching Name Server
  - Not authoritative for any zone, only caches entries for other zones.

Resource Records (RR)

- Each zone file contains a set of resource record for that zone.
- Each RR has: [Name, Type, TTL, Rdata, …]
- RR Types (16 bit value):
  - SOA : Start of Authority
  - NS: Authoritative name server for the domain
  - A: Hostname
  - MX: Mail Server
  - CNAME: Alias name
  - HINFO: CPU and OS info
  - PTR: Pointer to another part of the namespace
  - SRV: Service name (RFC 2782)
  - And some others….
- TTL: indicates how long the RR can be cached (32 bit integer in seconds)
- Rdata: a type specific value (for ex: an IP address for A type etc.)

Example Zone File


```
$ORIGIN example.com. ; used to make other entries FQDN
$TTL 3D ; For caching (typically in seconds)
@       IN      SOA     dns1.example.com. hostmaster.example.com. (
                        2001062501 ; serial
                        21600      ; refresh after 6 hours
                        3600       ; retry after 1 hour
                        604800     ; expire after 1 week
                        86400 )    ; minimum TTL of 1 day
        IN      NS      dns1.example.com.
        IN      NS      dns2.example.com.
        IN      MX      10      mail.example.com.
        IN      MX      20      mail2.example.com.
```

Name servers for this domain

Mail Server : here mail sent to a particular namespace controlled by this zone should go.

dns1	IN	A	10.0.1.1
dns2	IN	A	10.0.1.2
server1	IN	A	10.0.1.5
server2	IN	A	10.0.1.6
ftp	IN	A	10.0.1.3
	IN	A	10.0.1.4
mail	IN	CNAME	server1
mail2	IN	CNAME	server2
www	IN	CNAME	server1

Address record, which specifies an IP address to assign to a name

Canonical Name record, which maps one name to another.

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 is similar otherwise.

Reverse Zone File


```
$ORIGIN 1.0.10.in-addr.arpa.
$TTL 86400
@       IN      SOA     dns1.example.com.      hostmaster.example.com. (
                        2001062501 ; serial
                        21600      ; refresh after 6 hours
                        3600       ; retry after 1 hour
                        604800     ; expire after 1 week
                        86400 )    ; minimum TTL of 1 day

1       IN      PTR     dns1.example.com.
2       IN      PTR     dns2.example.com.
3       IN      PTR     ftp.example.com.
4       IN      PTR     ftp.example.com.
5       IN      PTR     server1.example.com.
6       IN      PTR     server2.example.com.
```

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.

Other protocol details

- Usually runs on UDP port 53
- Uses TCP for zone transfers (an some large responses)
- TCP can also be used for normal operation, though not used normally.
- Same message format for query and response.

DNS Query Example


```
Domain Name System (query)
    [Response In: 1852]
    Transaction ID: 0x241a
    Flags: 0x0100 (Standard query)
        0... .... .... .... = Response: Message is a query
        .000 0... .... .... = Opcode: Standard query (0)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... .0.. .... = Z: reserved (0)
        .... .... ...0 .... = Non-authenticated data OK: Non-authenticated data is unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.google.com: type A, class IN
            Name: www.google.com
            Type: A (Host address)
            Class: IN (0x0001)
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00      ........ ......E.
0010  00 3c 51 e3 40 00 40 11  ea cb 7f 00 00 01 7f 00      .<Q.@.@. ........
0020  00 01 ec ed 00 35 00 28  fe 3b 24 1a 01 00 00 01      .....5.( .;$.....
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c      .......w ww.googl
0040  65 03 63 6f 6d 00 00 01  00 01                       e.com... ..
```

DNS Response

Domain Name System (response)

[Request In: 1851]

[Time: 0.000125000 seconds]

Transaction ID: 0x241a

Flags: 0x8180 (Standard query response, No error)

1... .... = Response: Message is a response

.000 0... .... = Opcode: Standard query (0)

.... .0.. .... = Authoritative: Server is not an authority for domain

.... ..0. .... = Truncated: Message is not truncated

....1 .... = Recursion desired: Do query recursively

.... 1... .... = Recursion available: Server can do recursive queries

.... .0.. .... = Z: reserved (0)

....0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server

.... 0000 = Reply code: No error (0)

Questions: 1

Answer RRs: 3

Authority RRs: 0

Additional RRs: 0

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E.
0010  00 7a 00 00 40 00 40 11  3c 71 7f 00 00 01 7f 00   .z..@.@. <q......
0020  00 01 00 35 ec ed 00 66  fe 79 24 1a 81 80 00 01   ...5...f .y$.....
0030  00 03 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   .......w ww.googl
0040  65 03 63 6f 6d 00 00 01  00 01 c0 0c 00 05 00 01   e.com... ........
0050  00 05 28 39 00 12 03 77  77 77 01 6c 06 67 6f 6f   ..(9...w ww.l.goo
0060  67 6c 65 03 63 6f 6d 00  c0 2c 00 01 00 01 00 00   gle.com. .,......
0070  00 e3 00 04 42 f9 59 63  c0 2c 00 01 00 01 00 00   ....B.Yc .,......
0080  00 e3 00 04 42 f9 59 68                           ....B.Yh
```

Queries

www.google.com: type A, class IN

Name: www.google.com

Type: A (Host address)

Class: IN (0x0001)

Answers

www.google.com: type CNAME, class IN, cname www.l.google.com

Name: www.google.com

Type: CNAME (Canonical name for an alias)

Class: IN (0x0001)

Time to live: 3 days, 21 hours, 52 minutes, 57 seconds

Data length: 18

Primary name: www.l.google.com

www.l.google.com: type A, class IN, addr 66.249.89.99

Name: www.l.google.com

Type: A (Host address)

Class: IN (0x0001)

Time to live: 3 minutes, 47 seconds

Data length: 4

Addr: 66.249.89.99

www.l.google.com: type A, class IN, addr 66.249.89.104

Name: www.l.google.com

Type: A (Host address)

Class: IN (0x0001)

Time to live: 3 minutes, 47 seconds

Data length: 4

Addr: 66.249.89.104