CS640: Computer Networks

Aditya Akella

Lecture 17
Naming and the DNS

Naming

- Need naming to identify resources

- Once identified, resource must be located

- How to name resource?
  - Naming hierarchy

- How do we efficiently locate resources?
  - DNS: name → location (IP address)

- Challenge: How do we scale these to the wide area?

2

/ETC/HOSTS

Why not use /etc/hosts?
- Original Name to Address Mapping
  - *Flat* namespace
  - Lookup mapping in /etc/hosts
  - SRI kept main copy
  - Downloaded regularly
- Count of hosts was increasing: machine per domain → machine per user
  - Many more downloads
  - Many more updates

3

Domain Name System Goals

- Basically a wide-area distributed database of name to IP mappings

- Goals:
  - Scalability
  - Decentralized maintenance
  - Robustness

4

DNS Records

- DB contains tuples called resource records (RRs)
  - Classes = Internet (IN), Chaosnet (CH), etc.
  - Each class defines value associated with type

RR format: **(class, name, value, type, ttl)**

FOR IN class:

- Type=A
  - **name** is hostname
  - **value** is IP address
- Type=NS
  - **name** is domain (e.g. foo.com)
  - **value** is name of authoritative name server for this domain

- Type=CNAME
  - **name** is an alias name for some "canonical" (the real) name
  - **value** is canonical name
- Type=MX
  - **value** is hostname of mailserver associated with **name**

Properties of DNS Host Entries

- Different kinds of mappings are possible:
  - Simple case: 1-1 mapping between domain name and IP addr:
    - `kittyhawk.cmcl.cs.cmu.edu` maps to `128.2.194.242`
  - Multiple domain names maps to the same IP address:
    - `eecs.mit.edu` and `cs.mit.edu` both map to `18.62.1.6`
  - Single domain name maps to multiple IP addresses:
    - `aol.com` and `www.aol.com` map to multiple IP addrs.

6

2

Programmer's View of DNS

- Conceptually, programmers can view the DNS database as a collection of millions of *host entry structures*:

```
/* DNS host entry structure */
struct hostent {
    char      *h_name;            /* official domain name of host */
    char      **h_aliases;       /* null-terminated array of domain names */
    int       h_addrtype;        /* host address type (AF_INET) */
    int       h_length;          /* length of an address, in bytes */
    char      **h_addr_list;     /* null-terminated array of in_addr structs */
};
```
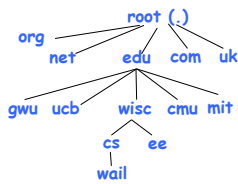
  – in_addr is a struct consisting of 4-byte IP address
- Functions for retrieving host entries from DNS:
  – `gethostbyname`: query key is a DNS host name.
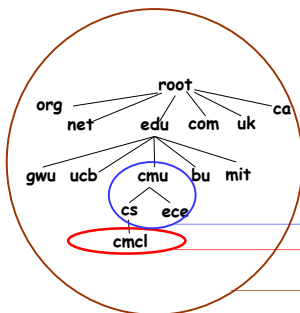 – `gethostbyaddr`: query key is an IP address.

7

DNS Design: Hierarchy Definitions

- Each node in hierarchy stores a list of names that end with same suffix
  - Suffix = path up tree

- E.g., given this tree, where would following be stored:
  - Fred.com
  - Fred.edu
  - Fred.wisc.edu
  - Fred.cs.wisc.edu
  - Fred.cs.cmu.edu

root (.)

org
net   edu   com   uk
gwu ucb   wisc cmu mit
       cs   ee
       wail

8

DNS Design: Zone Definitions

- Zone = contiguous section of name space
  - E.g., Complete tree, single node or subtree

- A zone has an associated set of **name servers**
  - Knows mappings for zone

root
org   ca
net   edu   com   uk
gwu ucb   cmu bu mit
       cs   ece
       cmcl

→ Subtree
→ Single node
→ Complete Tree

9

3

DNS Design: Cont.

- Zones are created by convincing owner node to create/delegate a subzone
  - Records within zone store multiple redundant name servers
  - Primary/master name server updated manually
  - Secondary/redundant servers updated by zone transfer of name space
    - Zone transfer is a bulk transfer of the "configuration" of a DNS server – uses TCP to ensure reliability
  - The owner node creates an NS record for the sub-zone
    - Points to the name server for the new sub-zone
- Example:
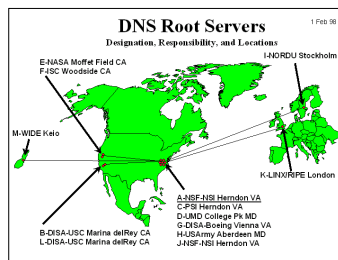 - CS.WISC.EDU created by WISC.EDU administrators

10

Servers/Resolvers

- Each host has a resolver
  - Typically a library that applications can link to
  - Resolver contacts name server
  - Local name servers hand-configured (e.g. /etc/resolv.conf)
- Name servers
  - Either responsible for some zone
    - Has mappings for all names in zone
  - Or knows of name servers for sub-zones
    - These servers know better about names in sub-zones
  - Names for which no mapping is known, direct requestor to root

11

DNS: Root Name Servers

- Responsible for "root" zone

- Approx. 13 root name servers worldwide
  - Currently {a-m}.root-servers.net

- Local name servers contact root servers when they cannot resolve a name
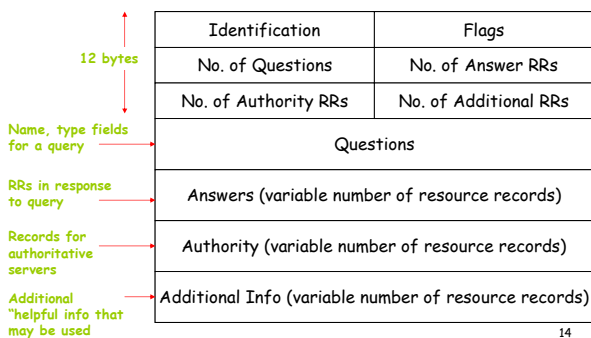 - Configured with well-known root servers

DNS Root Servers
Designation, Responsibility, and Locations   1 Feb 98

E-NASA Moffet Field CA
F-ISC Woodside CA
I-NORDU Stockholm
M-WIDE Keio
K-LINX/RIPE London
A-NSF-NSI Herndon VA
C-PSI Herndon VA
D-UMD College Pk MD
G-DISA-Boeing Vienna VA
H-USArmy Aberdeen MD
J-NSF-NSI Herndon VA
B-DISA-USC Marina delRey CA
L-DISA-USC Marina delRey CA

12

4

Typical Resolution

- Steps for resolving www.wisc.edu
  - Application calls gethostbyname() (RESOLVER)
  - Resolver contacts local name server (S_1)
  - S_1 queries root server (S_2) for (www.wisc.edu)
  - S_2 returns NS record for wisc.edu (S_3)
  - What about A record for S_3?
    - This is what the additional information section is for (PREFETCHING)
  - S_1 queries S_3 for www.wisc.edu
  - S_3 returns A record for www.wisc.edu

- Can return multiple A records → what does this mean?

13

DNS Message Format

Identification	Flags
No. of Questions	No. of Answer RRs
No. of Authority RRs	No. of Additional RRs
Questions	
Answers (variable number of resource records)	
Authority (variable number of resource records)	
Additional Info (variable number of resource records)	

12 bytes

Name, type fields for a query

RRs in response to query

Records for authoritative servers

Additional "helpful info that may be used

14

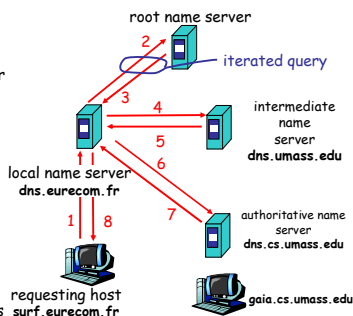## Lookup Methods

Recursive query:
- Server goes out and searches for more info (recursive)
- Only returns final answer or "not found"

Iterative query:
- Server responds with as much as it knows (iterative)
- "I don't know this name, but ask this server"

Workload impact on choice?
- Local server typically does recursive
- Root/distant server does iterative

root name server

iterated query

intermediate name server
dns.umass.edu

local name server
dns.eurecom.fr

authoritative name server
dns.cs.umass.edu

requesting host
surf.eurecom.fr

gaia.cs.umass.edu

1 2 3 4 5 6 7 8

15

Workload and Caching

- Are all servers/names likely to be equally popular?
  - Why might this be a problem? How can we solve this problem?

- DNS responses are cached
  - Quick response for repeated translations
  - Other queries may reuse some parts of lookup
    - NS records for domains

- DNS negative queries are cached
  - Don't have to repeat past mistakes
  - E.g. misspellings, search strings in resolv.conf

- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
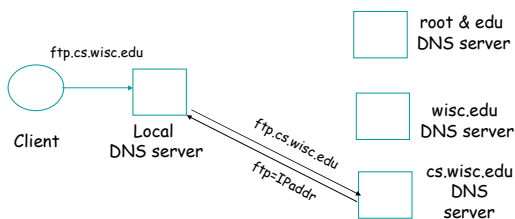 - TTL passed with every record

16

## Typical Resolution
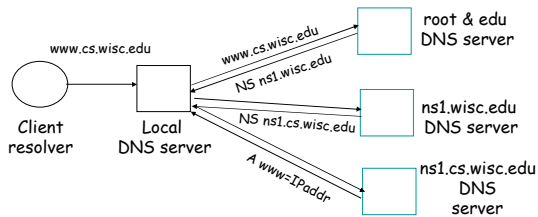
17

Subsequent Lookup Example

18

Reliability

- DNS servers are replicated
  - Name service available if ≥ one replica is up
  - Queries can be load balanced between replicas

- UDP used for queries
  - Need reliability → must implement this on top of UDP!
  - Why not just use TCP?

- Try alternate servers on timeout
  - Exponential backoff when retrying same server

- Same identifier for all queries
  - Don't care which server responds

19

Reverse DNS

unnamed root
arpa edu
in-addr cmu
128 cs
2
194 cmcl
242 kittyhawk
128.2.194.242
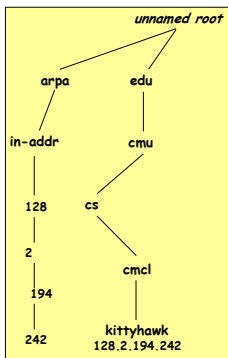
- Task
  - Given IP address, find its name
  - When is this needed?

- Method
  - Maintain separate hierarchy based on IP names
  - Write 128.2.194.242 as 242.194.2.128.in-addr.arpa
    - Why is the address reversed?

- Managing
  - Authority manages IP addresses assigned to it
  - E.g., CMU manages name space 2.128.in-addr.arpa

20

Prefetching

- Name servers can add additional data to response

- Typically used for prefetching
  - CNAME/MX/NS typically point to another host name
  - Responses include address of host referred to in "additional section"

21

New gTLDs

- .info → general info
- .biz → businesses
- .aero → air-transport industry
- .coop → business cooperatives
- .name → individuals
- .pro → accountants, lawyers, and physicians
- .museum → museums
- Only new one actives so far = .info, .biz, .name

22