CS640: Computer Networks

Aditya Akella

Lecture 17
Naming and the DNS

The Road Ahead

- DNS Design
- DNS Today

2

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?

3

Obvious Solutions (1)

Lookup a Central DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Single point of update
- Doesn't *scale!*

4

Obvious Solutions (2)

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

5

Domain Name System Goals

- Basically a wide-area distributed database of name to IP mappings
- Goals:
  - Scalability
  - Decentralized maintenance
  - Robustness
  - Global scope
    - Names mean the same thing everywhere
  - Don't need
    - Atomicity
    - Strong consistency

6

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 Message Format

12 bytes	Identification	Flags
	No. of Questions	No. of Answer RRs
	No. of Authority RRs	No. of Additional RRs
Name, type fields for a query	Questions (variable number of answers)	
RRs in response to query	Answers (variable number of resource records)	
Records for authoritative servers	Authority (variable number of resource records)	
Additional "helpful info that may be used	Additional Info (variable number of resource records)	

8

DNS Header Fields

- Identification
  - Used to match up request/response
- Flags
  - 1-bit to mark query or response
  - 1-bit to mark authoritative or not
  - 1-bit to request recursive resolution
  - 1-bit to indicate support for recursive resolution

9

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:

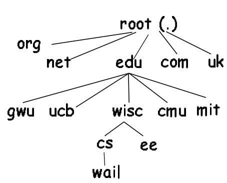
- Type=A
  - **name** is hostname
  - **value** is IP address
- Type=NS
  - **name** is domain (e.g. foo.com)
  - **value** is name of authoritative name server for this domain
- Type=CNAME
  - **name** is an alias name for some "canonical" (the real) name
  - **value** is canonical name
- Type=MX
  - **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.
  - Some valid domain names don't map to any IP address:
    - for example: `cs.wisc.edu`

11

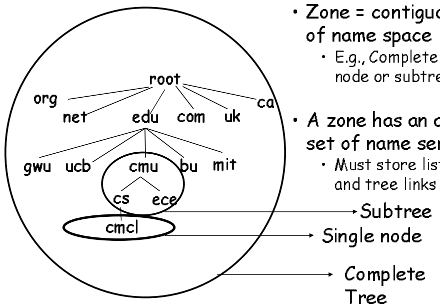
DNS Design: Hierarchy Definitions



- 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

12

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
  - Must store list of names and tree links

13

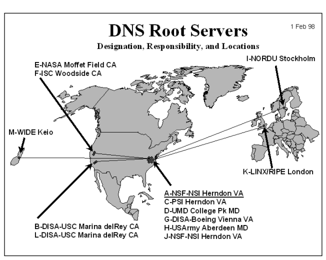
DNS Design: Cont.

- 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
- Example:
  - CS.WISC.EDU created by WISC.EDU administrators
  - Who creates WISC.EDU or .EDU?

14

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



15

Servers/Resolvers

- Each host has a resolver
  - Typically a library that applications can link to
  - Resolves contacts name server
  - Local name servers hand-configured (e.g. /etc/resolv.conf)
- Name servers
  - Either responsible for some zone or…
  - Local servers
    - Do lookup of distant host names for local hosts
    - Typically answer queries about local zone

16

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?

17

Lookup Methods

Recursive query:

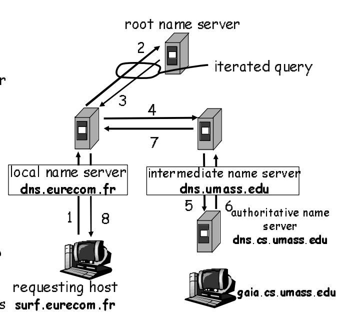
- Server goes out and searches for more info (recursive)
- Only returns final answer or "not found"

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



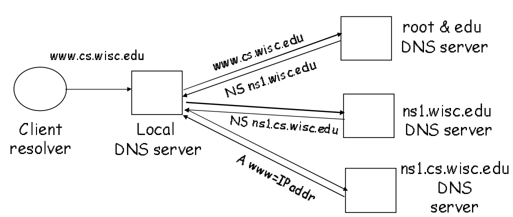
18

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

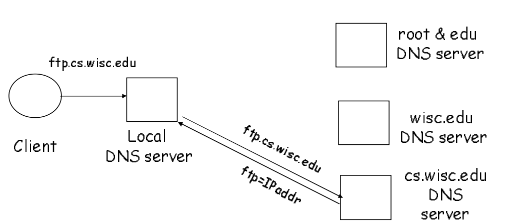
19

Typical Resolution



20

Subsequent Lookup Example



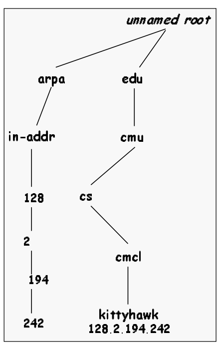
21

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

22

Reverse DNS



- 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

23

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"

24

DNS Today: Root Zone

- Generic Top Level Domains (gTLD) = .com, .net, .org, etc…
- Country Code Top Level Domain (ccTLD) = .us, .ca, .fi, .uk, etc…
- Root server ({a-m}.root-servers.net) also used to cover gTLD domains
  - Load on root servers was growing quickly!
  - Moving .com, .net, .org off root servers was clearly necessary to reduce load → done Aug 2000

25

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

26

DNS Performance

- No centralized caching per site
  - Each machine runs own caching local server
  - Why is this a problem?
  - How many hosts do we need to share cache? → recent studies suggest 10-20 hosts
- Hit rate for DNS = 80% → 1 - (#DNS/#connections)
  - Is this good or bad?
- Most Internet traffic is Web
  - What does a typical page look like? → average of 4-5 imbedded objects → needs 4-5 transfers
  - This alone accounts for 80% hit rate!
- Lower TTLs for A records does not affect performance
- DNS performance really relies more on NS-record caching

27

Summary

- Motivations → large distributed database
  - Scalability
  - Independent update
  - Robustness
- Hierarchical database structure
  - Zones
  - How is a lookup done
- Caching/prefetching and TTLs
- Reverse name lookup

28