

15-441 Lecture 7

DNS

Copyright © Seth Goldstein, 2008

Based on slides from previous 441 lectures

15-441

© Seth Goldstein

1

Outline

- DNS Design
- DNS Today

(Extra credit, remind me at end)

15-441

© Seth Goldstein

2

What is DNS?

- DNS (Domain Name Service) is primarily used to translate human readable names into machine usable addresses, e.g., IP addresses.
- DNS goal:
 - Efficiently locate resources.
E.g., Map name → IP address
 - Scale to many users over a large area
 - Scale to many updates

15-441

© Seth Goldstein

3

How resolve name → IP addr?

15-441

© Seth Goldstein

4

Obvious Solutions (1)

Why not centralize DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Single point of update

- Doesn't *scale!*

LECTURE 14

CHAPTER 9: DNS

11

Obvious Solutions (2)

Why not use */etc/hosts*?

- Original Name to Address Mapping
 - Flat namespace
 - */etc/hosts*
 - SRI kept main copy
 - Downloaded regularly
- Mid 80's this became untenable. Why?
- Count of hosts was increasing: machine per domain → machine per user
 - Many more downloads
 - Many more updates

/etc/hosts still exists.

LECTURE 14

CHAPTER 9: DNS

12

LECTURE 14

CHAPTER 9: DNS

13

Domain Name System Goals

- Basically a wide-area distributed database
(The biggest in the world!)
- Scalability
- Decentralized maintenance
- Robustness
- Global scope
 - Names mean the same thing everywhere
- Don't need all of ACID
 - Atomicity
 - Strong consistency
- Do need: distributed update/query & Performance

LECTURE 14

CHAPTER 9: DNS

13

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 */
};
```

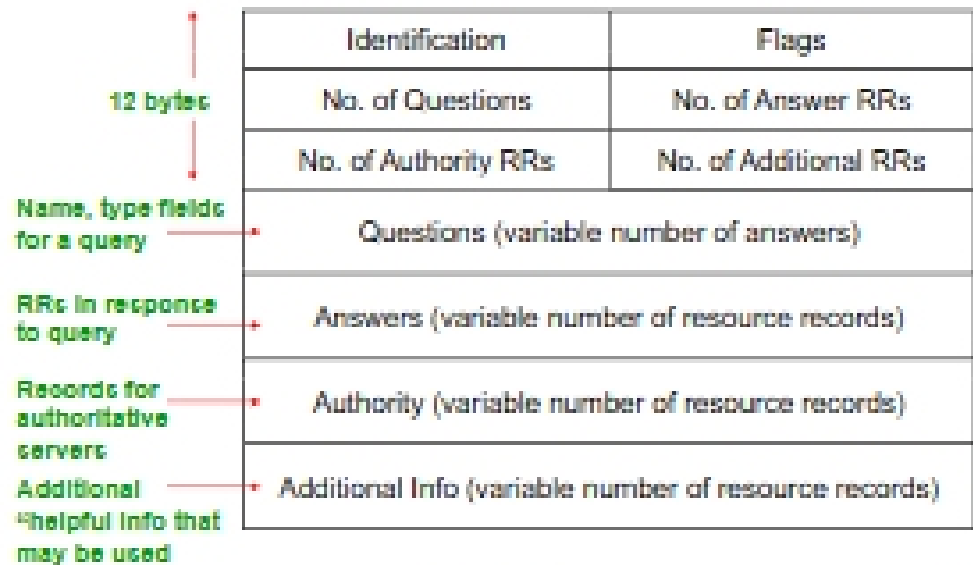
- *in_addr* is a struct consisting of 4-byte IP addr
- Functions for retrieving host entries from DNS:
 - *gethostbyname*: query key is a DNS host name.
 - *gethostbyaddr*: query key is an IP address.

LECTURE 14

CHAPTER 9: DNS

14

DNS Message Format



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

DNS Records

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

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

For "IN" class:

<ul style="list-style-type: none"> • Type=A <ul style="list-style-type: none"> - name is hostname - value is IP address 	<ul style="list-style-type: none"> • Type=CNAME <ul style="list-style-type: none"> - name is an alias name for same "canonical" name - value is canonical name
<ul style="list-style-type: none"> • Type=NS <ul style="list-style-type: none"> - name is domain (e.g. foo.com) - value is name of authoritative name server for this domain 	<ul style="list-style-type: none"> • Type=MX <ul style="list-style-type: none"> - value is hostname of mailserver associated with name

Properties of DNS Host Entries

- Different kinds of mappings are possible:
- 1-1 mapping between domain name and IP addr:
 - provolve.crcl.cs.cmu.edu maps to 128.2.218.81
 - Multiple domain names maps to the same IP addr:
 - www.scs.cmu.edu and www.cs.cmu.edu both map to 128.2.203.164
 - 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 addr:
 - crcl.cs.cmu.edu doesn't have a host