Distributed Systems
分布式系统

Name Services
命名服务

for more details, see

http://www.zytrax.com/books/dns/

Name, address, route

- The name indicates what we seek
  - Human readable
  - Universal Resource Names (URNs)
- An address indicates where it is
  - IP address, port
- A route tells how to get there
  - Internet routing

Name and IP-address Not 1 to 1 Mapping

- One host may map to more than one name
  - One server machine may be the web server (www.foo.com), mail server (mail.foo.com), etc.
- One host may have more than one IP address
  - IP addresses are per network interface
  - A machine may have multiple network interfaces (e.g., a gateway)
- Names don't necessarily reflect geographical locations

Name Hierarchy

- Naming in Internet is Hierarchical
  - Better scalability (decreasing centralization)
  - Better name space management
- Example: weather.yahoo.com belongs to yahoo.com which belongs to .com
  - Regulated by global non-profit organizations
- First, get a domain name; then you are free to assign subnames in that domain
  - How to get a domain name (see later)

Top-level Domains

- Country Code Domains
  - .uk, .de, .jp, .us, .tv,...
- Generic Domains
  - .aero, .biz, .com, .coop, .edu, .gov, .info, .int, .mil, .museum, .name, .net, .org, and .pro
- Infrastructure Domain (**A**ddress and **R**outing **P**arameter **A**rea domain)
  - .arpa

How to get a domain name?

- The Internet Corporation for Assigned Names and Numbers (ICANN) is an internationally organized, non-profit corporation that has responsibility for
  - generic and country code Top-Level Domain name system management,
  - IP address space allocation,
  - protocol identifier assignment, and
  - root server system management functions
- ICANN authorizes other companies to register domains

Name Services

- What is the name service: A name server maintains a database of bindings between human-readable names and attributes of objects (locations, addresses, etc).
- Why difficult: In open distributed systems, name database is distributed and managed by different servers, and those servers are required to cooperate to resolve names.
- Requirements:
  - openness
  - scalability
  - fault tolerance (availability)

DNS: Domain Name System

- Distributed database implemented in hierarchy of many name servers
- DNS services:
  - host name resolution
  - mail host location (e.g., find the mail server for hwdu@hitsz.edu.cn)
  - reverse resolution
  - well-known services (e.g., telnet, FTP, HTTP, etc)
- The add/delete of a name is done by an authoritative administrator manually editing the name database.

DNS Name Servers

Why not centralize DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance

It doesn't *scale!*

DNS based on RFC 1034 by Mockapertris in 1987:

- fully distributed and hierarchical structure
- no server has all name-to-IP address mappings

Name server:

- a process running on a host that processes DNS requests

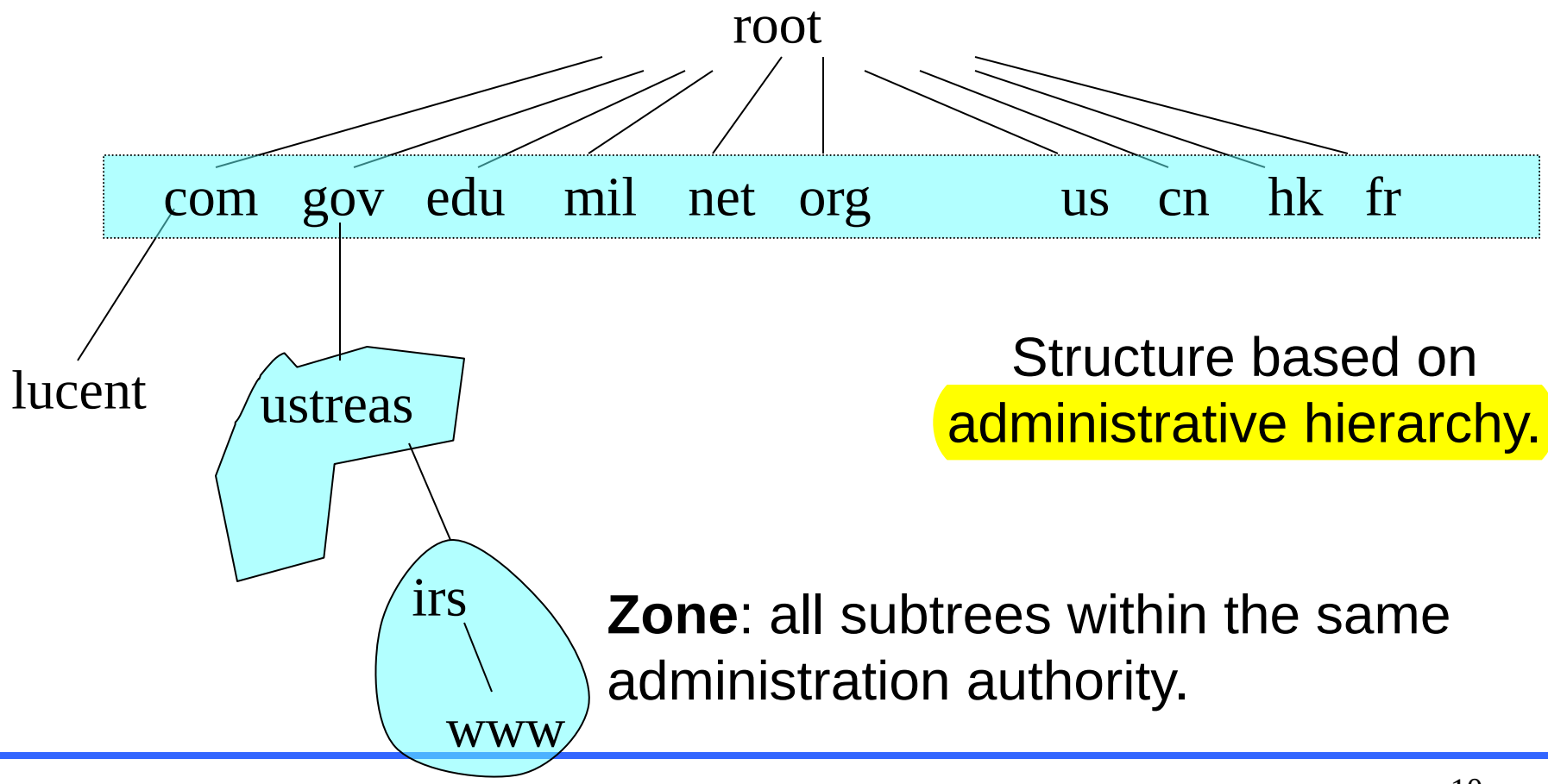
Local name server:

- each ISP, company, department, has a *local name server*
- DNS queries always first go to local name server

Authoritative name server:

- perform name/address translation for a specific domain or zone
- database for name / address mapping are maintained by administrators
- name servers are organized in a hierarchical structure as hierarchy of the name space.

Name Server Zone Structure

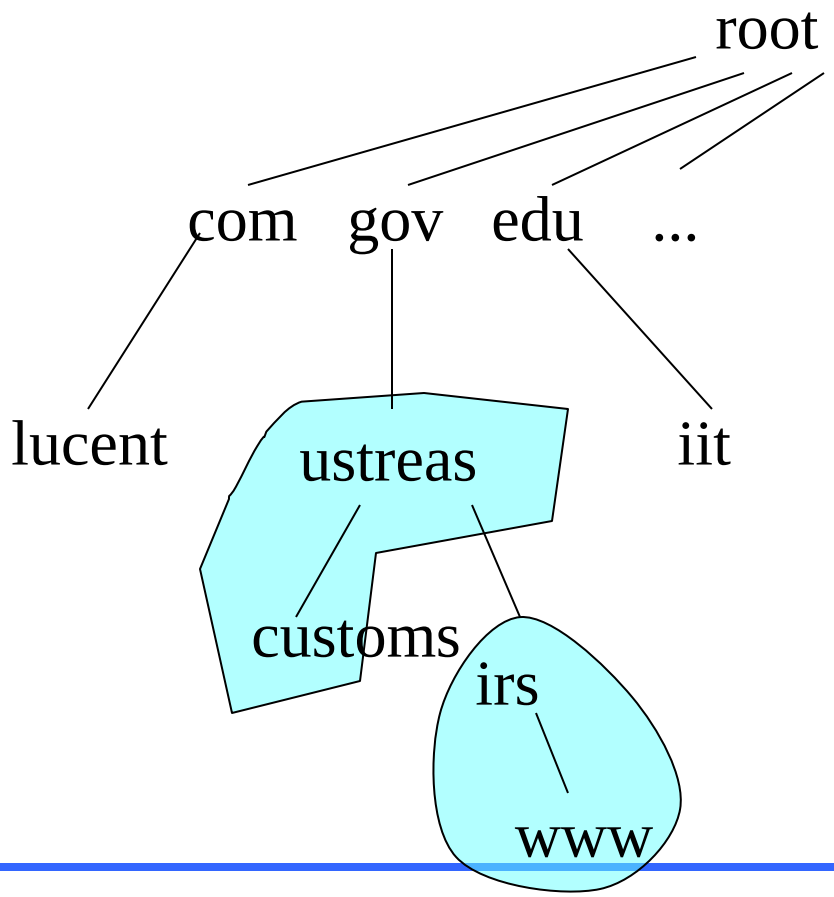


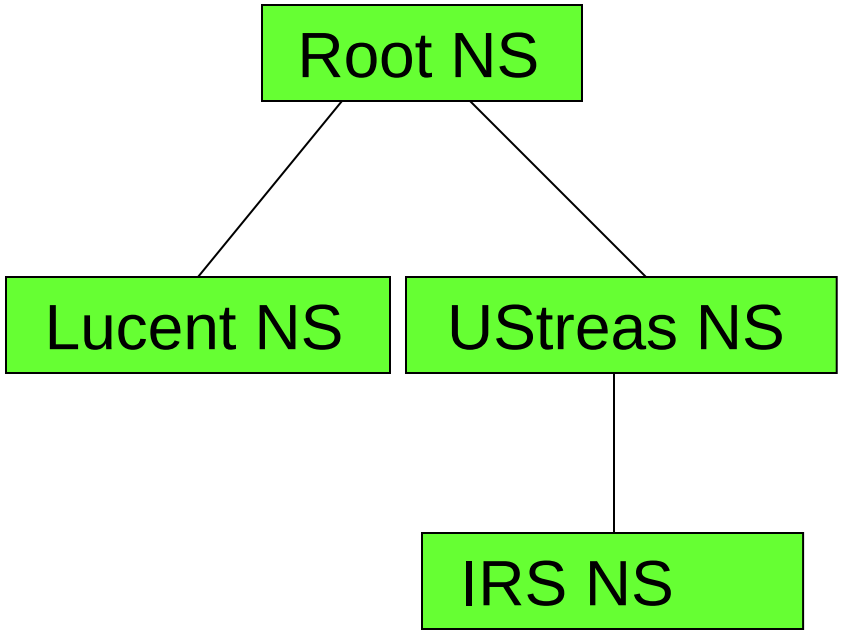
Structure based on administrative hierarchy.

Zone: all subtrees within the same administration authority.

Name Servers (NS)

root zone db: http://www.iana.org/domains/root/db



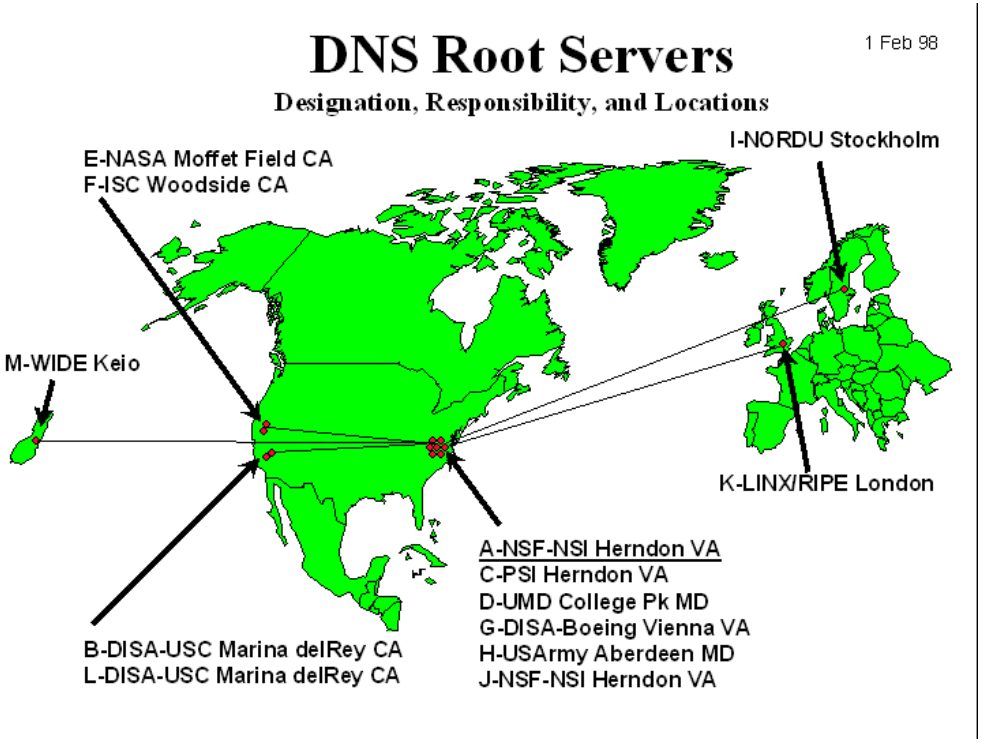


Name Servers (NS)

- NSs are **duplicated** for reliability. Each domain must have a primary and a secondary name server.
- Each host knows the IP address of the **local** NS.
- Each NS knows the IP addresses of root NSs. A query is forwarded directly to the root NS if it cannot be resolved by a local name server.
- Information of root NSs (downloaded to local NS as *root.cache*) at http://www.internic.net/zones/named.root

DNS: Root name servers

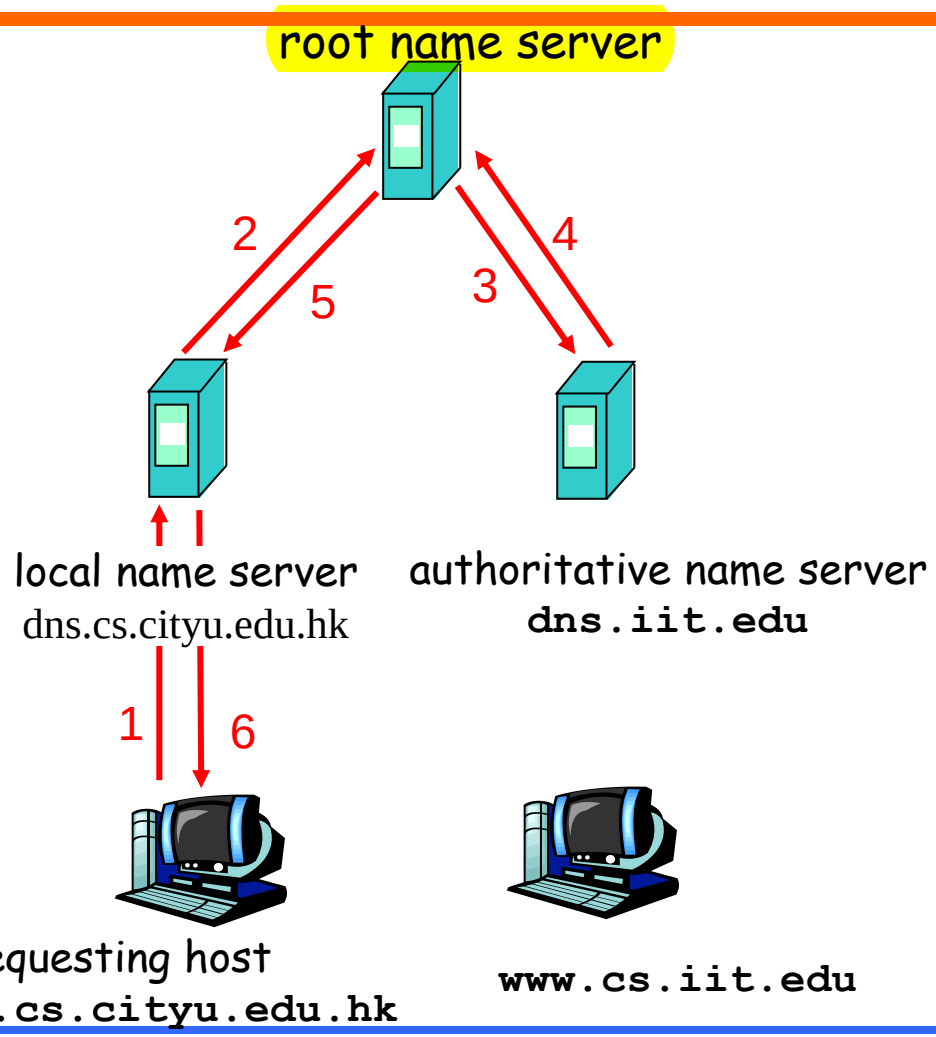
- Root NS is contacted by local name servers when they cannot resolve names
- It knows the authoritative name servers at the top level



Simple DNS example (recursive query)

host **`cslab.cs.cityu.edu.hk`**
wants IP address of
`www.cs.iit.edu`

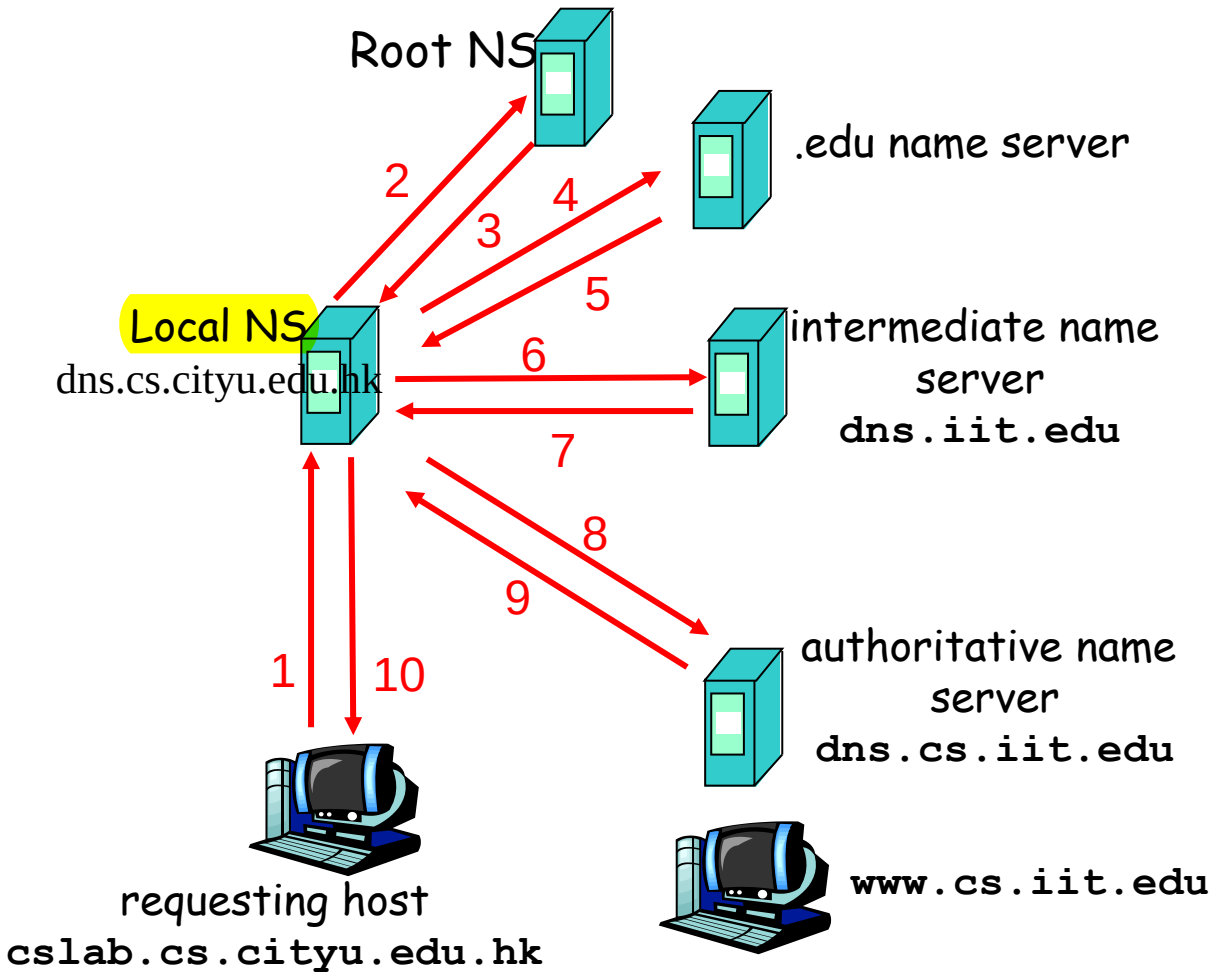
1. Contacts its local DNS server,
`dns.cs.cityu.edu.hk`
2. **`dns.cs.cityu.edu.hk`**
requests root name server, if necessary
3. root name server requests authoritative name server,
`dns.iit.edu`, if necessary



DNS example (iterative query)

Root name server:

- may not know the dest NS for a name
- but knows *intermediate NS:* who to contact to find the next level of dest NS



Recursive Query vs. Iterative Query

- DNS standard supports both recursive query and iterative query. When a DNS server makes a resolution-request, it needs to specify if it's recursive or iterative.
- Some DNS servers, mainly the high level servers, don't support recursive queries for security reason.
- To resolve a name, a mix of recursive and iterative queries may be used by DNS servers at different levels. A client (the user who makes the initial request to its local DNS server) usually use "recursive" query for convenience.

Primary server and Secondary server

For fault tolerance, there are two servers providing authoritative data in a domain:

primary server: the server reads the domain data directly from a local master file.

secondary server: it down loads the domain data from the primary server and communicates with the primary periodically to keep its data up-to-date.

Data Files for Name Servers

A name server (primary or secondary) uses 3 data files for name resolutions:

- name resolution file: names to IP addresses
  - e.g. /var/named/*db.cs.cityu.edu.hk*
- reverse translation file: IP addresses to names
  - e.g. /var/named/*db.144.214.120*
- cached file: cached data from previous queries
  - e.g. /var/named/*db.cache*

DNS records: More than Name to IP Address

DNS: distributed db storing resource records (RR)

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

- Type=A
  - **`name`** is hostname
  - **`value`** is IP address

- Type=CNAME
  - **`name`** is an alias name for some "real name"
  - **`value`** is the "real name"

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

- Type=MX
  - **`value`** is hostname of mailserver associated with **`name`**

Other types

A	The Internet address of the host
CNAME	The canonical name for an alias
HINFO	The host CPU and operating system type
MD	The mail destination
MX	The mail exchanger
MB	The mailbox domain name
MG	The mail group member
MINFO	The mailbox or mail list information
NS	The name server
PTR	The host name if the query is in the form of an IP address; otherwise the pointer to other information
SOA	The domain's start-of-authority information
TXT	The text information
UINFO	The user information
WKS	The supported well-known services

root.cache

```
;
; BIND data file for initial cache data for root domain servers.
;
.                       99999999        IN      NS      i.root-servers.net.
                        99999999        IN      NS      a.root-servers.net.
                        99999999        IN      NS      b.root-servers.net.
                        99999999        IN      NS      c.root-servers.net.
                        99999999        IN      NS      d.root-servers.net.
                        99999999        IN      NS      e.root-servers.net.
                        99999999        IN      NS      f.root-servers.net.
                        99999999        IN      NS      g.root-servers.net.
                        99999999        IN      NS      h.root-servers.net.
                        99999999        IN      NS      j.root-servers.net.
                        99999999        IN      NS      k.root-servers.net.
                        99999999        IN      NS      l.root-servers.net.
                        99999999        IN      NS      m.root-servers.net.
a.root-servers.net.     99999999        IN      A       198.41.0.4
b.root-servers.net.     99999999        IN      A       192.228.79.201
c.root-servers.net.     99999999        IN      A       192.33.4.12
d.root-servers.net.     99999999        IN      A       128.8.10.90
e.root-servers.net.     99999999        IN      A       192.203.230.10
f.root-servers.net.     99999999        IN      A       192.5.5.241
g.root-servers.net.     99999999        IN      A       192.112.36.4
h.root-servers.net.     99999999        IN      A       128.63.2.53
i.root-servers.net.     99999999        IN      A       192.36.148.17
j.root-servers.net.     99999999        IN      A       192.58.128.30
k.root-servers.net.     99999999        IN      A       193.0.14.129
l.root-servers.net.     99999999        IN      A       198.32.64.12
m.root-servers.net.     99999999        IN      A       202.12.27.33
```

db.cs.cityu.edu.hk

```
;BIND DUMP V8
$ORIGIN cityu.edu.hk.
cs              3600            IN              SOA     mars.cs.cityu.edu.hk. root.mars.cs.cityu.edu.hk. (
                9505145 3600 600 604800 3600 )          ;Cl=4
                1200            IN              NS      ns1.cs.cityu.edu.hk.            ;Cl=4
                1200            IN              NS      ns2.cs.cityu.edu.hk.            ;Cl=4
                1200            IN              A       144.214.120.1 ;Cl=4
                1200            IN              MX      10 mail.cs.cityu.edu.hk.        ;Cl=4
$ORIGIN cs.cityu.edu.hk.
fts0            1200            IN              A       144.214.120.102                 ;Cl=4
sbh8            1200            IN              A       144.214.120.136                 ;Cl=4
cslec           3600            IN              A       144.214.120.26                  ;Cl=4
sbh9            1200            IN              A       144.214.120.137                 ;Cl=4
smtp            1200            IN              A       144.214.120.1 ;Cl=4
fts1            1200            IN              A       144.214.120.103                 ;Cl=4
www1            1200            IN              A       144.214.120.19                  ;Cl=4
fts2            1200            IN              A       144.214.120.100                 ;Cl=4
www2            1200            IN              A       144.214.120.6 ;Cl=4
ojserver        3600            IN              A       144.214.120.160                 ;Cl=4
cs3334          3600            IN              A       144.214.120.172                 ;Cl=4
fts3            1200            IN              A       144.214.120.101                 ;Cl=4
.....
```

db.144.214.120

```
;BIND DUMP V8
$ORIGIN 214.144.in-addr.arpa.
120   3600    IN      NS      mars.cs.cityu.edu.hk.          ;Cl=5
      3600    IN      NS      csl.cs.cityu.edu.hk.           ;Cl=5
      3600    IN      SOA     mars.cs.cityu.edu.hk. root.mars.cs.cityu.edu.hk. (
              9503328 3600 600 604800 3600 )          ;Cl=5
$ORIGIN 120.214.144.in-addr.arpa.
130   3600    IN      PTR     sbh2.cs.cityu.edu.hk.          ;Cl=5
129   3600    IN      PTR     sbh1.cs.cityu.edu.hk.          ;Cl=5
131   3600    IN      PTR     sbh3.cs.cityu.edu.hk.          ;Cl=5
132   3600    IN      PTR     sbh4.cs.cityu.edu.hk.          ;Cl=5
133   3600    IN      PTR     sbh5.cs.cityu.edu.hk.          ;Cl=5
134   3600    IN      PTR     sbh6.cs.cityu.edu.hk.          ;Cl=5
1     3600    IN      PTR     mars.cs.cityu.edu.hk.          ;Cl=5
135   3600    IN      PTR     sbh7.cs.cityu.edu.hk.          ;Cl=5
136   3600    IN      PTR     sbh8.cs.cityu.edu.hk.          ;Cl=5
3     3600    IN      PTR     www.cs.cityu.edu.hk.           ;Cl=5
137   3600    IN      PTR     sbh9.cs.cityu.edu.hk.          ;Cl=5
4     3600    IN      PTR     web.cs.cityu.edu.hk.           ;Cl=5
140   3600    IN      PTR     sbh12.cs.cityu.edu.hk.         ;Cl=5
138   3600    IN      PTR     sbh10.cs.cityu.edu.hk.         ;Cl=5
5     3600    IN      PTR     gateway.cs.cityu.edu.hk.       ;Cl=5
.....
```

Primary Server Configuration

The primary server is configured by a local file (the file at the site of the primary server) */etc/named.boot*. At boot-up time, it reads this file.

```
// file name: named.boot
directory                                   /etc
primary     cs.cityu.edu.hk                 db.cs.cityu.edu.hk
primary     120.214.144.IN-ADDR.ARPA        db.120.214.144
cache       .                               db.cache
```

Secondary Server Configuration

The secondary server is configured by a local file (the file at the site of the secondary server) */etc/named.boot*. At boot-up time, it reads this file.

```
directory                                   /etc
secondary       cs.cityu.edu.hk  144.214.120.97    db.snd
.........
cache           .                           db.cache
```

Client Node Configuration in DNS

Each client computer in the domain has a file /etc/resolv.conf containing the addr of the local domain NS.

```
domain cs.cityu.edu.hk
nameserver 144.214.121.221
nameserver 144.214.121.220
```

Administration Operations on DNS

- add a new host into the domain: adding a record in the NS DB

  jupiter IN A 144.214.120.2

- create a sub-domain "*ds*": adding a record in the NS DB (and an entry for host name "ds-sun0")

  IN NS ds-sun0.**ds**.cs.cityu.edu.hk

- set the mail server for sub-domain "*ds*": adding a record

  ds.cs.cityu.edu.hk IN MX 1 mars.cs.cityu.edu.hk

  the "ds" sub-domain uses the same mail server as "cs.cityu" domain

- change www server for the domain: adding a record

  www.cs.cityu.edu.hk IN CNAME mars.cs.cityu.edu.hk

  OR

  www IN A 144.214.120.97

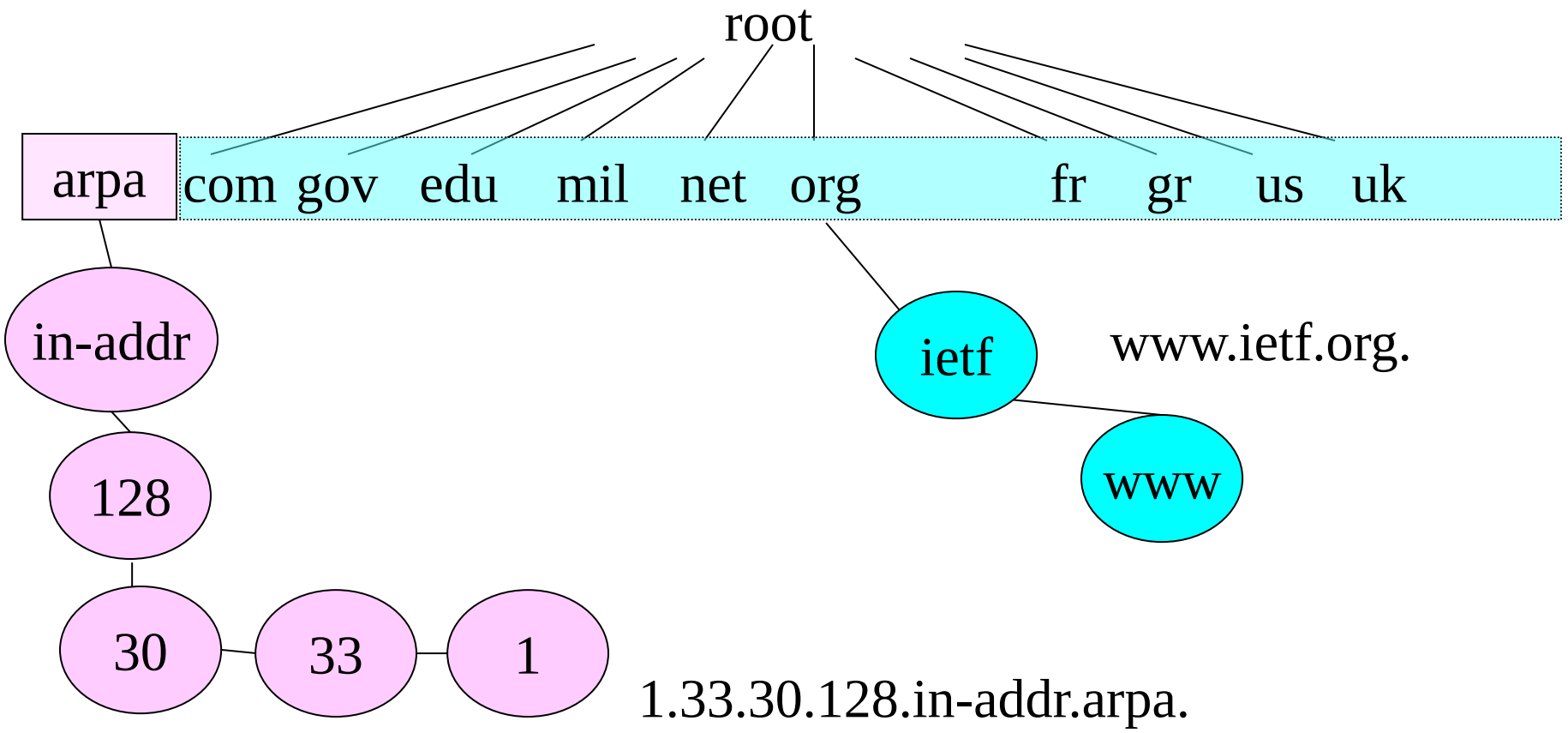
IP-address to Name: Reverse Mapping

What's the domain name for 128.30.33.1?

- Why is that hard? Which name server is responsible for that mapping? How do you find them?
- Answer: special root domain, ".in-addr.arpa", for reverse lookups

Top level domain .arpa

Want to know domain name for 128.30.33.1?
Issue a PTR request for 1.33.30.128.in-addr.arpa



Why resolution backwards?

For example: resolve 144.214.120.97 (demo at: http://www.dnsstuff.com/)

1) Ask *h.root-servers.net.* for 97.120.214.144.in-addr.arpa PTR record. It points to *figwort.arin.net.* (zone: 144.in-addr.arpa.)
2) Ask *figwort.arin.net.* for 97.120.214.144.in-addr.arpa PTR record. It points to *cpccspc.cityu.edu.hk.* (zone: 214.144.in-addr.arpa.)
3) Ask *cpccspc.cityu.edu.hk.* (NS for cityu) for 97.120.214.144.in-addr.arpa PTR record. It points to *mars.cs.cityu.edu.hk.* (zone: 120.214.144.in-addr.arpa.)
4) Ask *mars.cs.cityu.edu.hk.* (NS for cs.cityu) for 97.120.214.144.in-addr.arpa PTR record. It reports sbh16.cs.cityu.edu.hk.

in-addr.arpa domain

- When an organization acquires a domain name, it receives authority over the assigned domain name space. It can further assign the name space in more low levels.
- When an organization acquires a block of IP address space, it receives authority over the .in-addr.arpa address space.
- The domain name is always associated with IP block (network) address. Example: Acquire domain berkeley.edu and acquire a class B IP Network ID 128.143

DNS interface to clients

```
struct hostent *gethostbyname(const char *name);
struct hostent *gethostbyaddr(const char *addr, int len, int type);

struct hostent {
        char   *h_name;         /* canonical name of host */
        char   **h_aliases;     /* alias list */
        int     h_addrtype;     /* host address type */
        int     h_length;       /* length of address */
        char   **h_addr_list;   /* list of addresses */
};
```

DNS interface to clients

- resolver routines: make, send, and interpret queries and reply messages with Internet domain name servers:

  res_ninit, fp_resstat, res_hostalias, res_nquery,

  res_nsearch, res_nquerydomain, res_nmkquery, res_nsend, res_nclose, res_nsendsigned, dn_comp, dn_expand, hstrerror, res_init, res_query, res_search, res_mkquery, res_send, herror

  >> man resolver

DNS interface to clients

nslookup - query name servers interactively

nslookup [- option]... host [server]

nslookup [- option]... - [server]

nslookup

Examples:

- nslookup www.yahoo.com
- nslookup www.yahoo.com dns.cs.iit.edu
  - specify which local nameserver to use
- nslookup –type=mx cs.iit.edu
  - specify record type