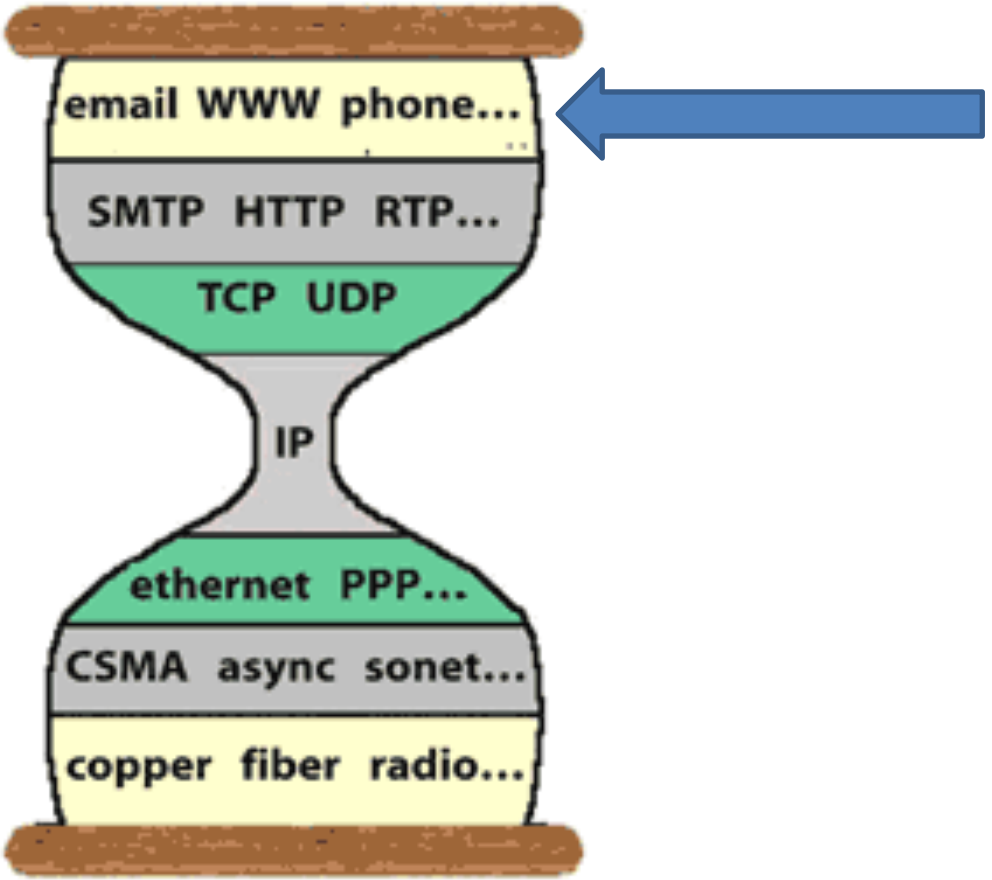
Plan for today: Domain Name System (DNS)

- Why we need it?
- What it is?
- How it works?
- What you can use it for?



Domain Name System

- Why Internet needs DNS
  - People: many identifiers (name, SSN, passport #)
  - Internet hosts, routers: also multiple IDs
    - IP address (32-bit) – used to deliver packets
    - Name (e.g., www.google.com) - used by humans
  - Domain Name System:
    - name → IP address translation
    - (for some IP) IP address → name
    - name → metadata (remember MX record from last lecture)

How people managed this before DNS?

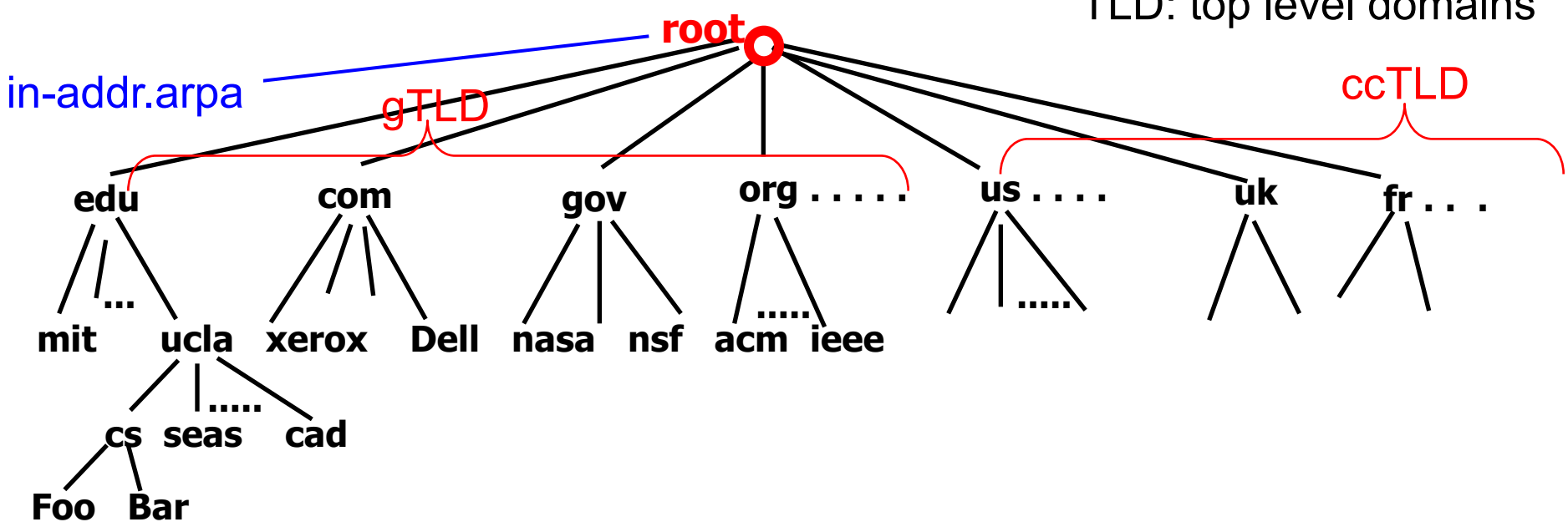
- DNS: following the same "query-reply" pattern as HTTP
- Normally runs on top UDP (unreliable transport)
  - can also run on top of TCP
- Security?
  - There are serious issues with DNS security

DNS: 4 Major Parts

important

- Defines a hierarchical name space
- Creates a distributed (federate) database, implemented through a hierarchy of authoritative servers
  - Provided by individual domain owners
- Caching resolvers look up the database
  - End hosts run a resolver routine (stub resolver)
  - Stub resolvers talk to the local caching resolvers provided by your Internet service provider
- Using DNS query protocol

DNS defines a hierarchical name space



- starting from the root, growing downward
  - variable depth
- each non-leaf node in the tree is a domain
  - Each domain belongs to an administrative authority
  - any domain can set up its own sub-domains, no limit on the depth
- DNS name hierarchy: independent from Internet's topological connectivity

Top-Level Domains

- Global TLDs (gTLD)
- Country code TLDs (ccTLD)
- I18n country code TLDs
  - .pф, .中国, …
- Generic TLDs
  - .kim, .bar, .coffee, .dance, .lol, + many more

- https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

Top-level Domain (TLD) Structure

- In 1983 (RFC 881), the idea was to have TLDs correspond to network service providers
  - e.g., ARPA, DDN, CSNET, etc.
    - Bad idea: if your network changes, your name changes
- By 1984 (RFC 920), functional domains was established
  - e.g., GOV for Government, COM for commercial, EDU for education, etc.
- RFC 920 also
  - Provided country domains
  - Provided “Multiorganizations”
    - Large, composed of other (particularly international) organizations
  - Provided a stable TLD structure until 1996 or so

Internet Corporation for Assigned Names and Numbers (ICANN)

- https://www.icann.org/
- Oversees the management of Internet resources including
  - Addresses
    - Delegating blocks of addresses to the regional registries
  - Protocol identifiers and parameters
    - Allocating port numbers, etc.
  - Names
    - Administration of the root zone file
    - Oversee the operation of the root name servers

The Root Nameservers

- The root zone file lists the names and IP addresses of the authoritative DNS servers for all top-level domains (TLDs)
- The root zone file is published on 13 servers, “A” through “M”, around the Internet
- Root name server operations currently provided by volunteer efforts by a very diverse set of organizations

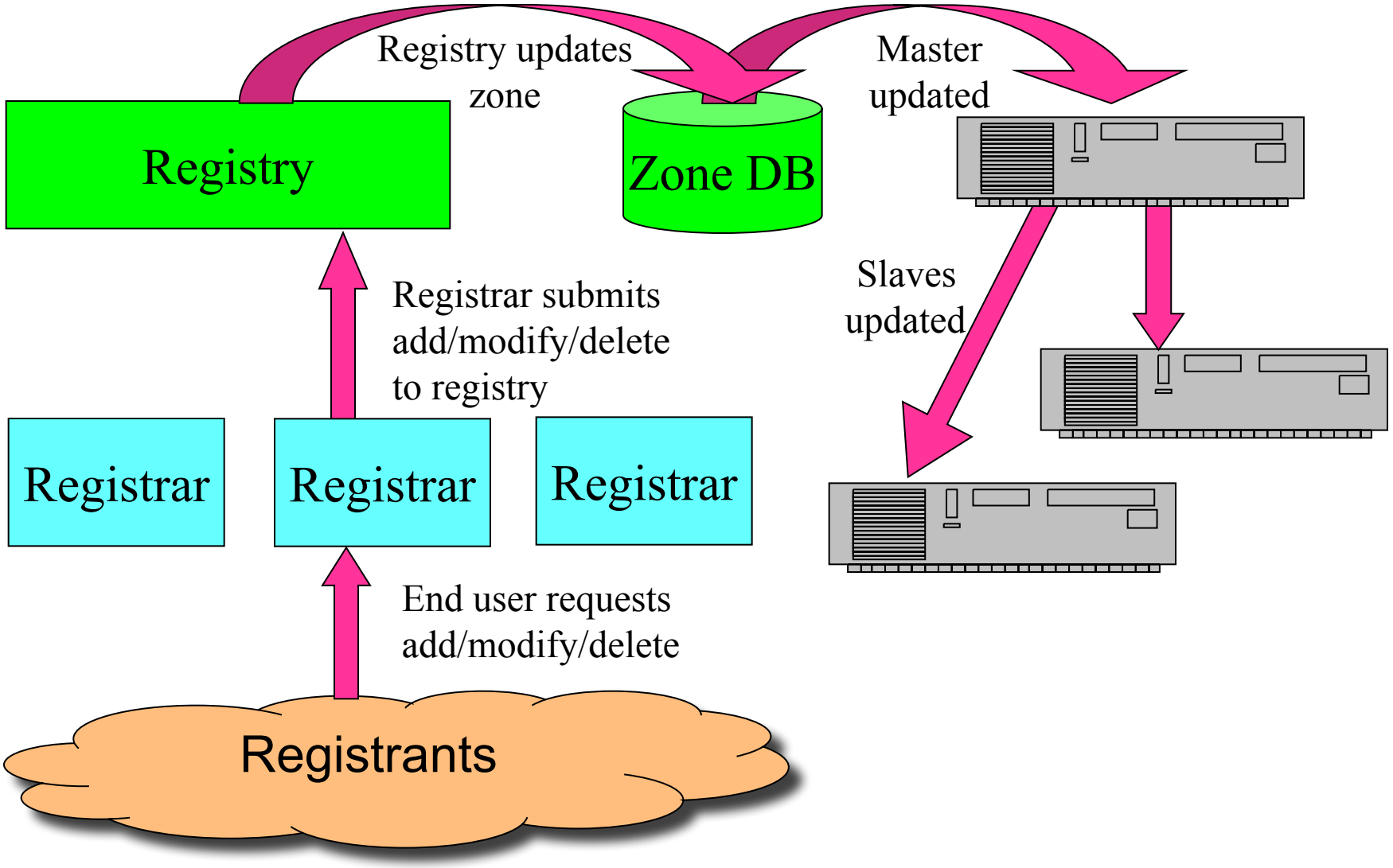
Root Name Server Operators

Nameserver	Operated by:
A	Verisign (US East Coast)
B	University of S. California –Information Sciences Institute (US West Coast)
C	Cogent Communications (US East Coast)
D	University of Maryland (US East Coast)
E	NASA (Ames) (US West Coast)
F	Internet Software Consortium (US West Coast)
G	U. S. Dept. of Defense (ARL) (US East Coast)
H	U. S. Dept. of Defense (DISA) (US East Coast)
I	Autonomica (SE)
J	Verisign (US East Coast)
K	RIPE-NCC (UK)
L	ICANN (US West Coast)
M	WIDE (JP)

Registries, Registrars, and Registrants

- A classification of roles in the operation of a domain name space
- Registry
  - the name space's database
  - the organization which has edit control of that database
  - the organization which runs the authoritative name servers for that name space
- Registrar
  - the agent which submits change requests to the registry on behalf of the registrant
- Registrant
  - the entity which makes use of the domain name

Registries, Registrars, and Registrants



Which Registrars You Know?

Verisign: the registry and registrar for gTLDs

- .COM, .NET, and .ORG
  - By far the largest top level domains on the Internet today

- Verisign received the contract for the registry for .COM, .NET, and .ORG
  - also a registrar for these TLDs

FYI

Reverse DND lookup: in-addr.arpa domain

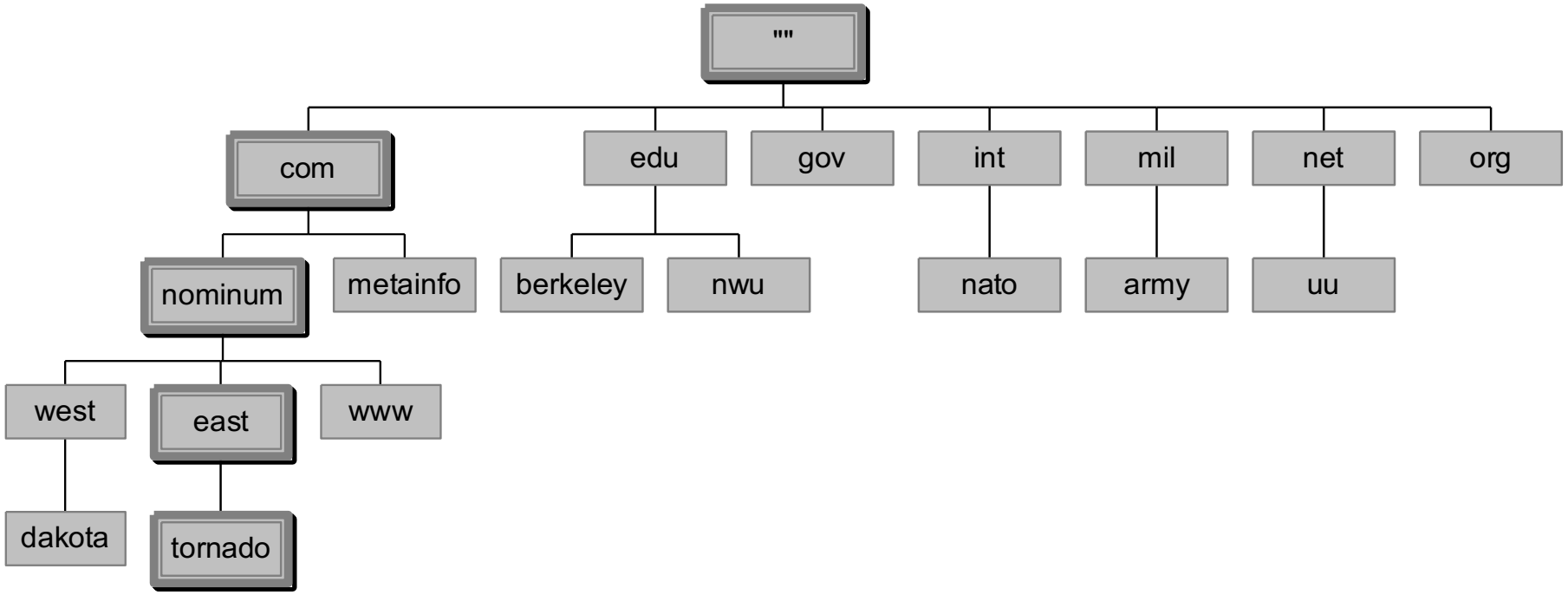
- Provides IP address to DNS name lookup
  - To get DNS name for address 1.2.3.4: query for 4.3.2.1.in-addr.arpa
- Why it is useful

```
traceroute to mit.edu (104.70.212.73), 64 hops max, 52 byte packets
 1  cisco196-1.cs.ucla.edu (131.179.196.3)  2.593 ms  2.289 ms  4.455 ms
 2  border1.cs.ucla.edu (131.179.244.3)  1.487 ms  1.365 ms  2.201 ms
 3  169.232.12.154 (169.232.12.154)  0.799 ms  0.991 ms  0.779 ms
 4  dr00f2.csb1--cr00f2.csb1.ucla.net (169.232.4.52)  0.839 ms  1.715 ms  1.037 ms
 5  cr00f2.csb1--bd11f1.anderson.ucla.net (169.232.4.5)  1.252 ms  1.266 ms  1.196 ms
 6  lax-agg6--ucla-10g.cenic.net (137.164.24.134)  1.971 ms  1.822 ms  1.756 ms
 7  206.111.14.81.ptr.us.xo.net (206.111.14.81)  2.074 ms  1.685 ms  1.659 ms
 8  207.88.14.212.ptr.us.xo.net (207.88.14.212)  23.648 ms  11.941 ms  30.415 ms
 9  207.88.13.25.ptr.us.xo.net (207.88.13.25)  2.063 ms  2.269 ms  1.793 ms
10  216.0.6.26 (216.0.6.26)  8.141 ms  8.528 ms  7.752 ms
11  0.ae0.pr0.lax00.tbone.rr.com (66.109.6.135)  3.338 ms
    0.ae4.pr0.lax00.tbone.rr.com (107.14.19.86)  1.661 ms
    0.ae2.pr0.lax00.tbone.rr.com (107.14.19.138)  1.877 ms
12  a104-70-212-73.deploy.static.akamaitechnologies.com (104.70.212.73)  1.698 ms  1.678 ms  1.681 ms
```

- Reverse DNS lookup is commonly used today as a *weak* security checking (e.g., by email servers)

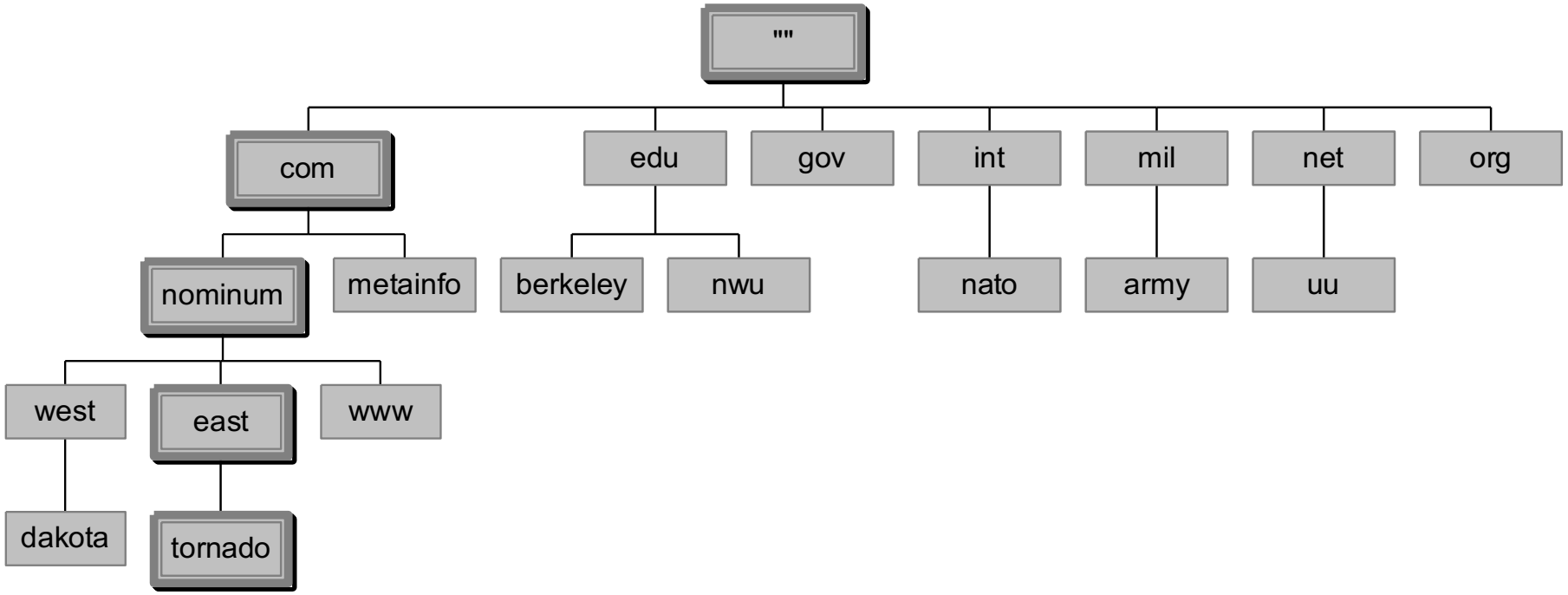
Domain Names

- A *domain name* is the sequence of labels from a node to the root, separated by dots (“.”s), read left to right
  - The name space has a maximum depth of 127 levels
  - Domain names are limited to 255 characters in length
- A node’s domain name identifies its position in the name space



Delegation

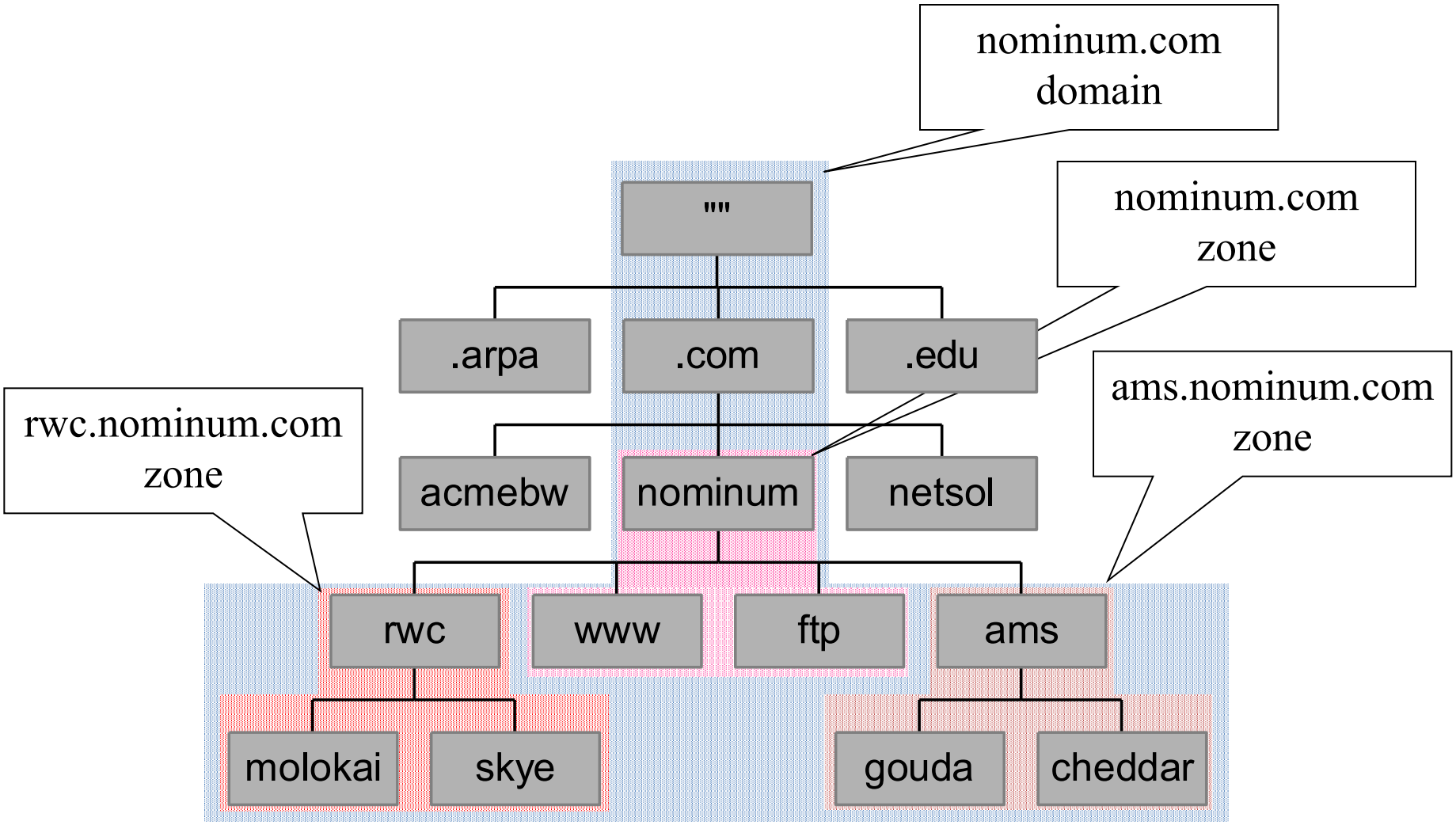
- Administrators can create subdomains to group hosts
  - According to geography, organizational affiliation etc.
- An administrator of a domain can delegate responsibility for managing a subdomain to someone else
- The parent domain retains links to the delegated subdomains



Delegation Creates DNS Zones

- Each time an administrator delegates a subdomain, a new unit of administration is created
  - The subdomain and its parent domain can now be administered independently
  - These units are called *zones*
  - The boundary between zones is a point of delegation in the name space
- Delegation is good: it is the key to scalability

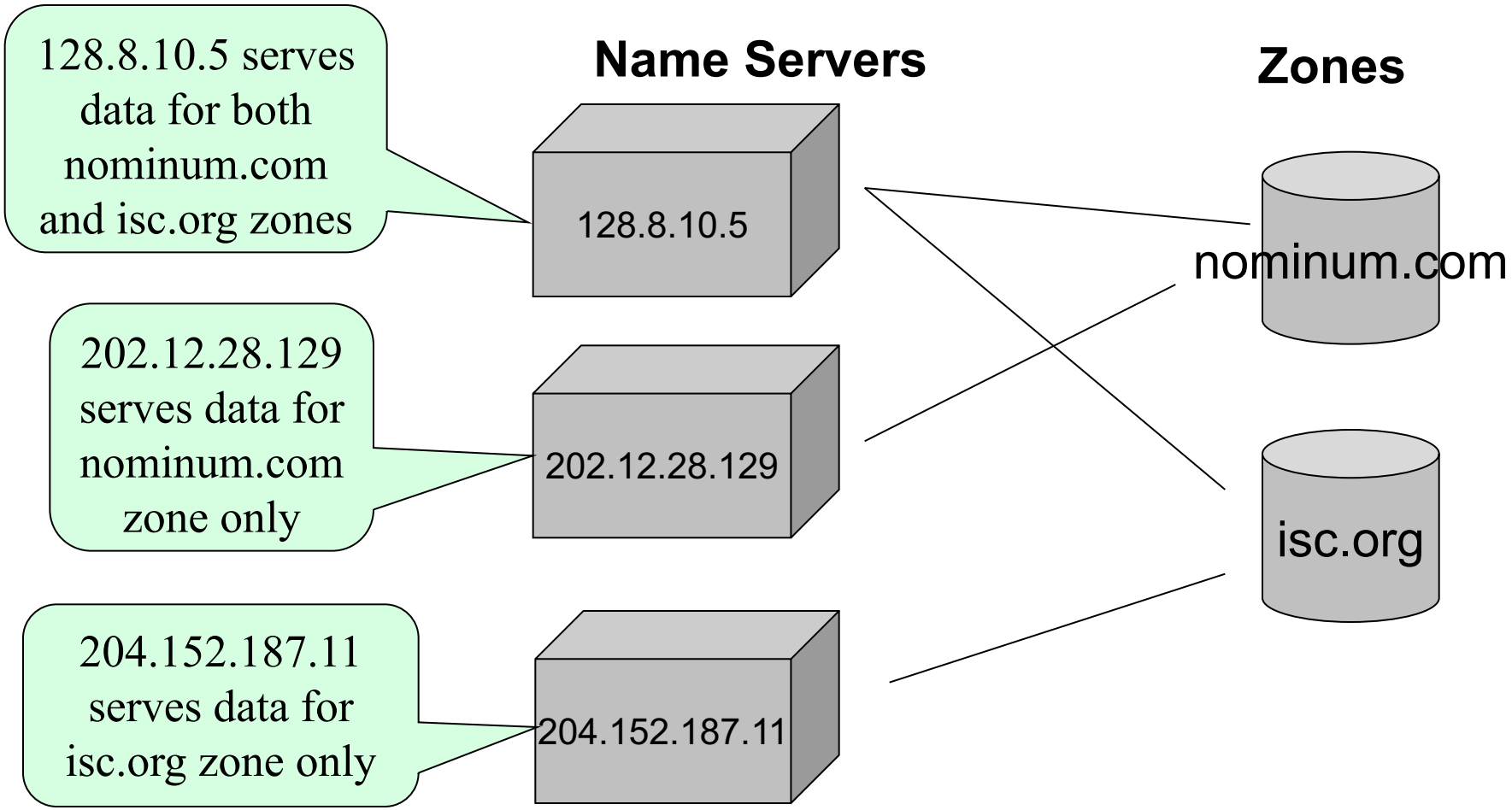
Dividing a Domain into Zones



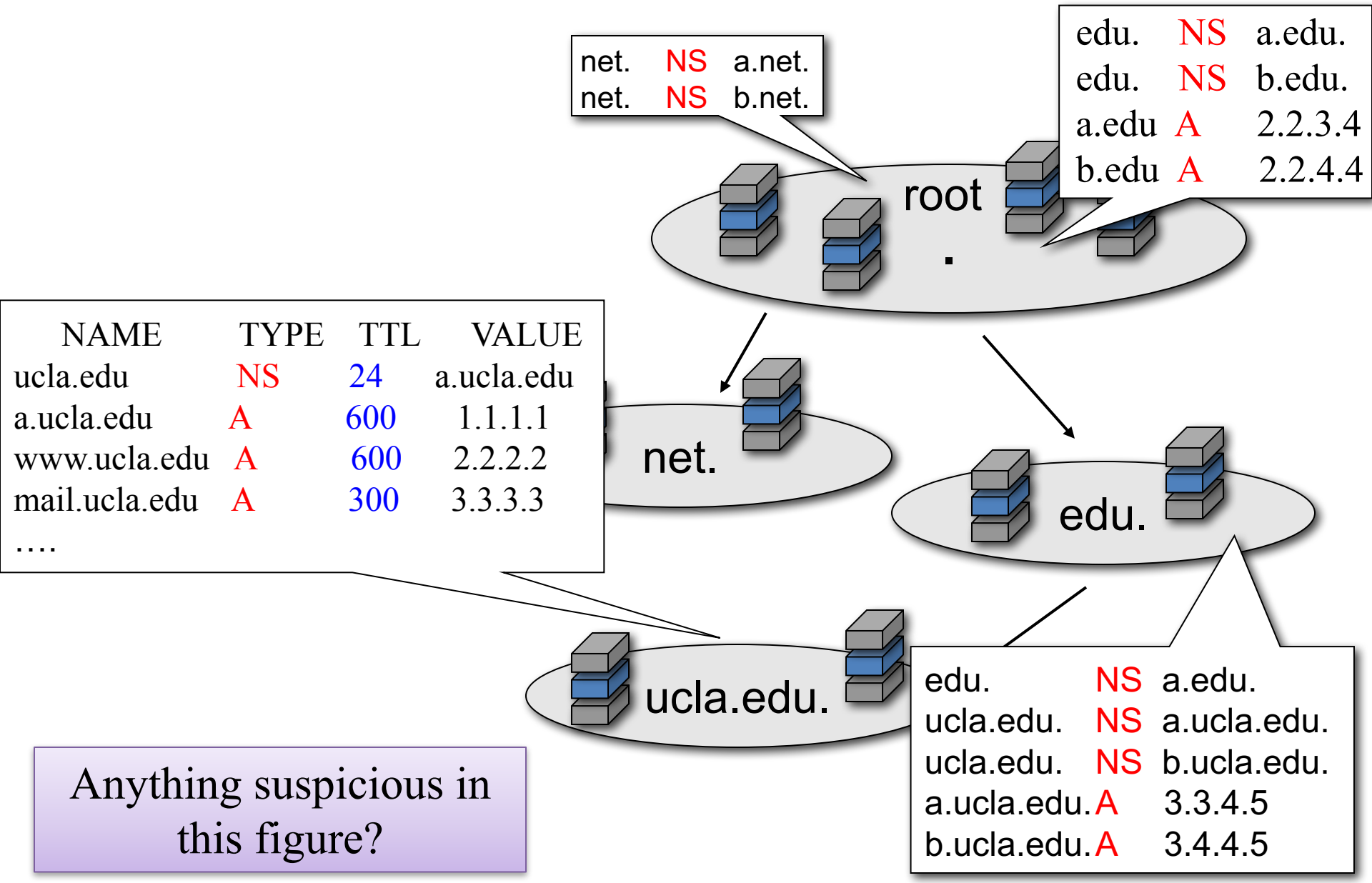
Name Servers

- Name servers store information about the name space in units called “zones”
  - The name servers that load a complete zone are said to “have authority for” or “be authoritative for” the zone
- Usually, more than one name server are authoritative for the same zone
  - This ensures redundancy and spreads the load
- Also, a single name server may be authoritative for many zones

Name Servers and Zones



Glue together all pieces of the distributed DB



Anything suspicious in this figure?

Didn’t Talk So Far

- Stub resolvers
- Caching name server

- Still remember what they are?

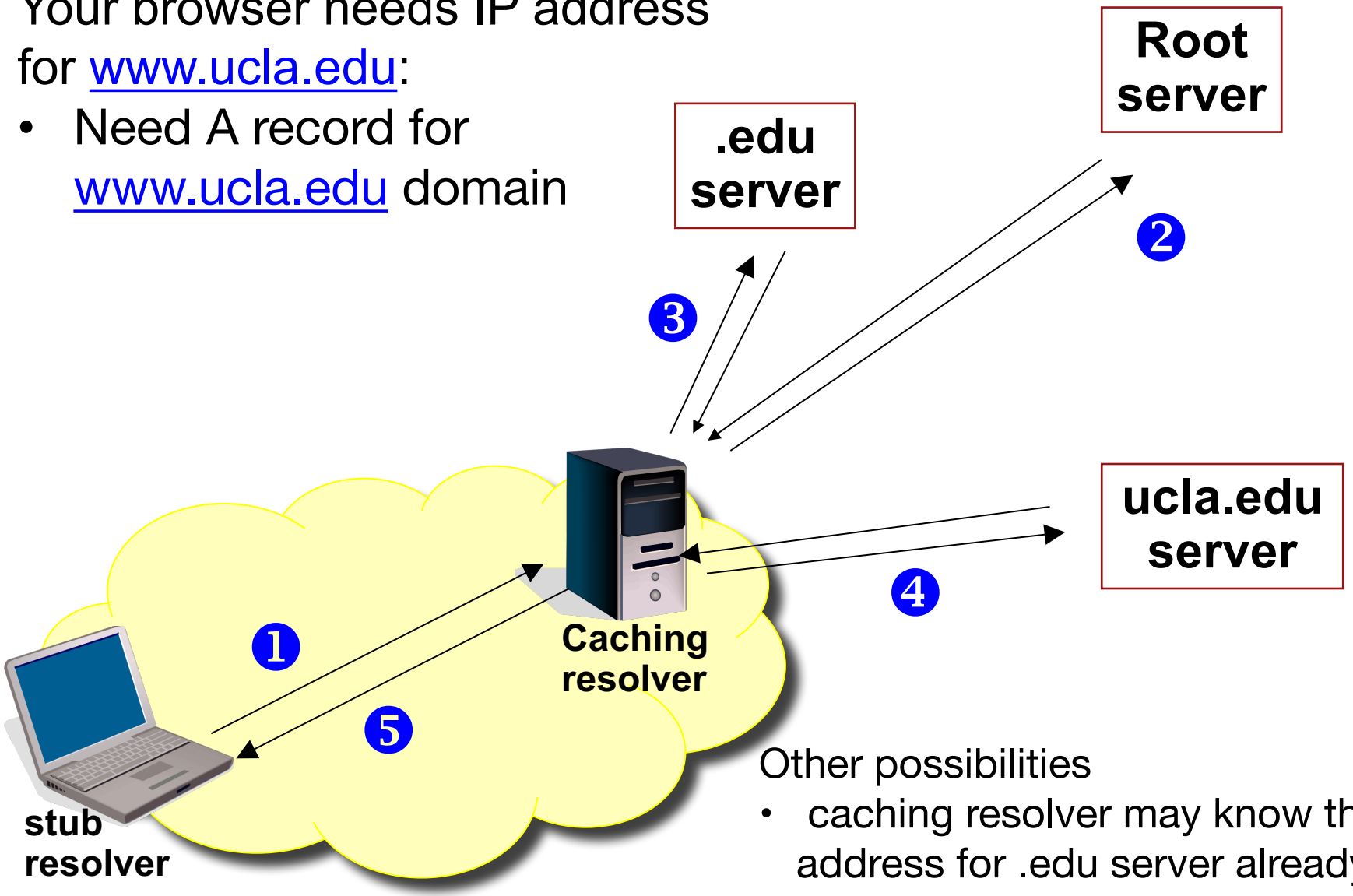
Types of Name Servers

- Authoritative
  - Maintains the data
    - Master – where the data is edited
    - Slave – where data is replicated to
- Caching
  - Stores data obtained from an authoritative server
- Stub resolvers configured with IP address of caching resolver(s) and send DNS queries to it (them)

Example of DNS Lookup

Your browser needs IP address for www.ucla.edu:

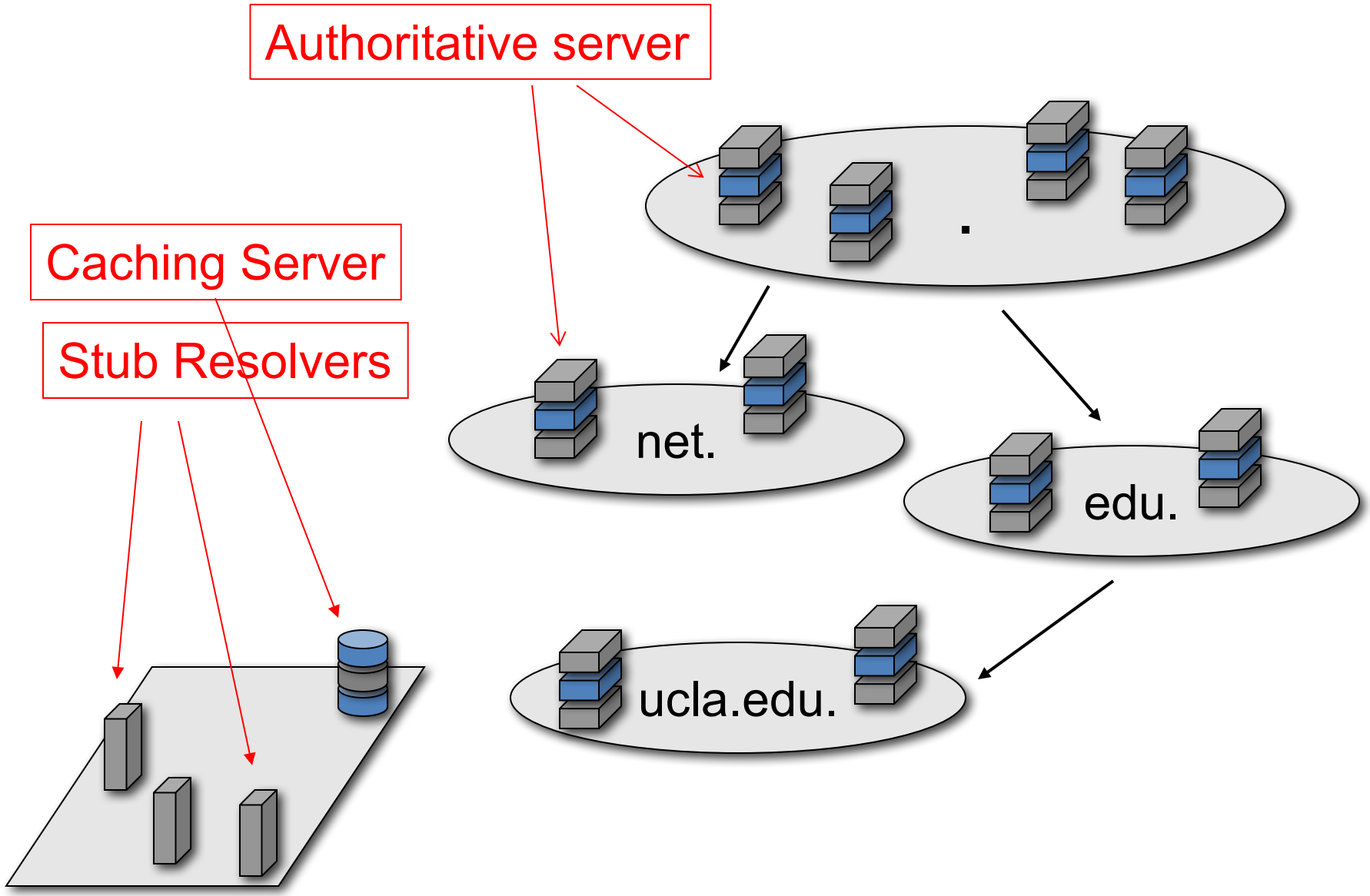
- Need A record for www.ucla.edu domain



Other possibilities

- caching resolver may know the IP address for .edu server already

Steps of Actions in Resolving a Name

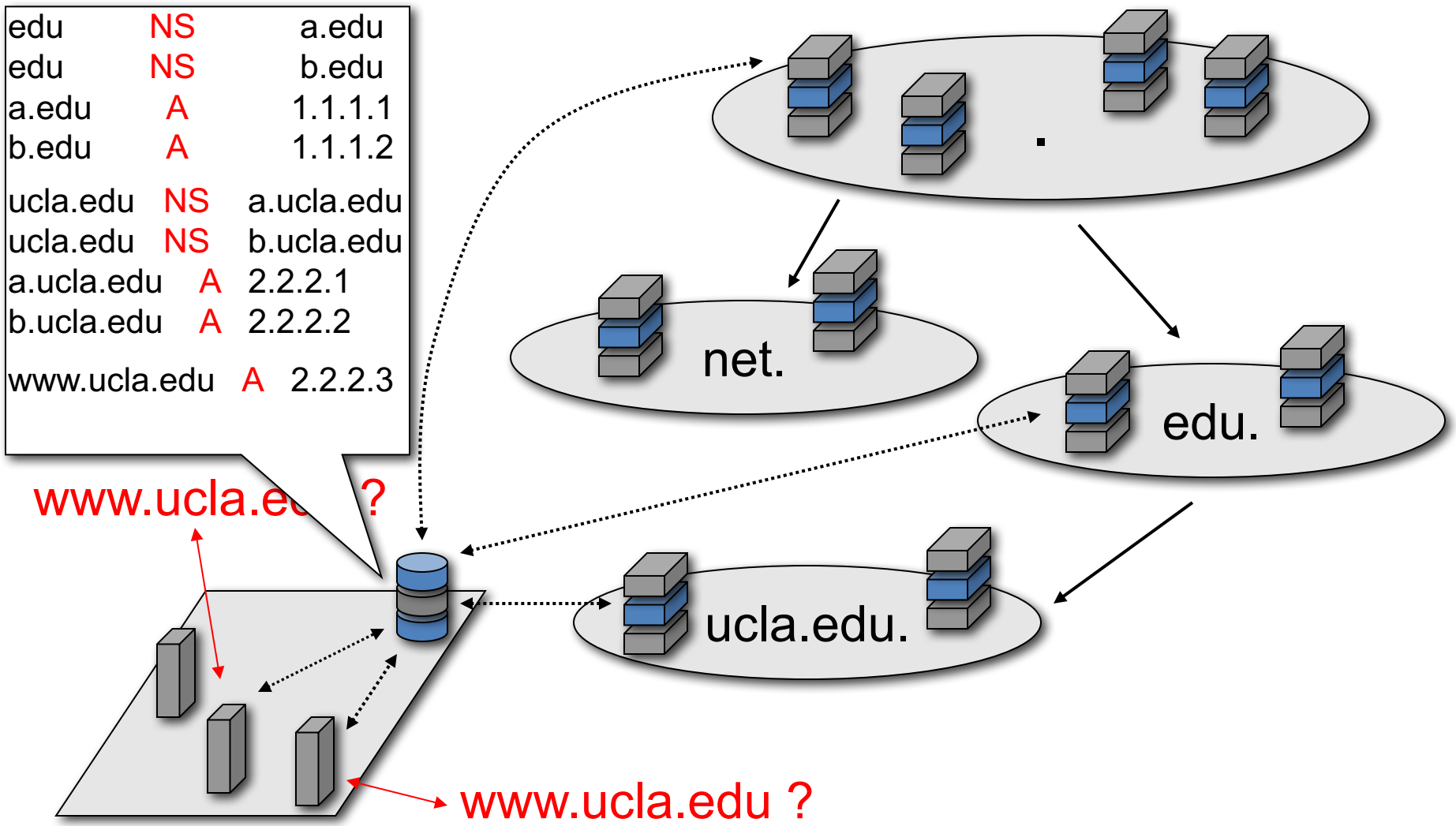


Steps of Actions in Resolving a Name

important

Cache

edu	NS	a.edu
edu	NS	b.edu
a.edu	A	1.1.1.1
b.edu	A	1.1.1.2
ucla.edu	NS	a.ucla.edu
ucla.edu	NS	b.ucla.edu
a.ucla.edu	A	2.2.2.1
b.ucla.edu	A	2.2.2.2
www.ucla.edu	A	2.2.2.3



Summary: How a DNS name gets resolved?

- Your host sends a query for www.ucla.edu (asking for IP address) to a local DNS caching resolver
  - provided by your ISP
- The caching resolver either finds a relevant answer in its cache, otherwise sends a query to one of the root servers
  - "relevant": any of the following
    - An exact match: www.ucla.edu's IP address
    - DNS server for ucla.edu
    - DNS server for .edu
- The root server replies with pointers to .edu servers
- The caching resolver queries .edu DNS server which replies with pointers to ucla.edu servers
- The caching resolver queries ucla.edu DNS server to get the IP address for www.ucla.edu, and sends the answer back to your host

Two ways of handling queries

- **recursive**: the server takes responsibility to get the final answer, then send back to the resolver

  Caching resolvers respond to recursive queries

  - Con's: the server has to do lots of the work!
  - Pro's: easy time for the resolver
- **iterative**: the server sends an answer back to the resolver directly

  Authority servers respond only to relevant iterative queries

  - Answer: either a matching reply, or a referral
  - Con's: the resolver has to do lots of work!
  - Pro'
    - easy job for the server
    - The resolver can build up a cache database

Bootstrapping DNS Resolution

- Stub resolvers
  - Need to send queries to at least one caching name server
- Caching name server / recursive resolvers
  - Need to send queries to unknown number of authoritative name
- How is it done in each case?

How to Provide Robust DNS Service

Very important!

- Robust service = high availability against network failure, server crash
- Two basic means: Redundancy & Caching
- Redundancy = replicated authoritative servers
- caching resolvers save a copy of all query replies
  - authoritative servers attach a TTL value to each reply
  - cache entries deleted after timeout

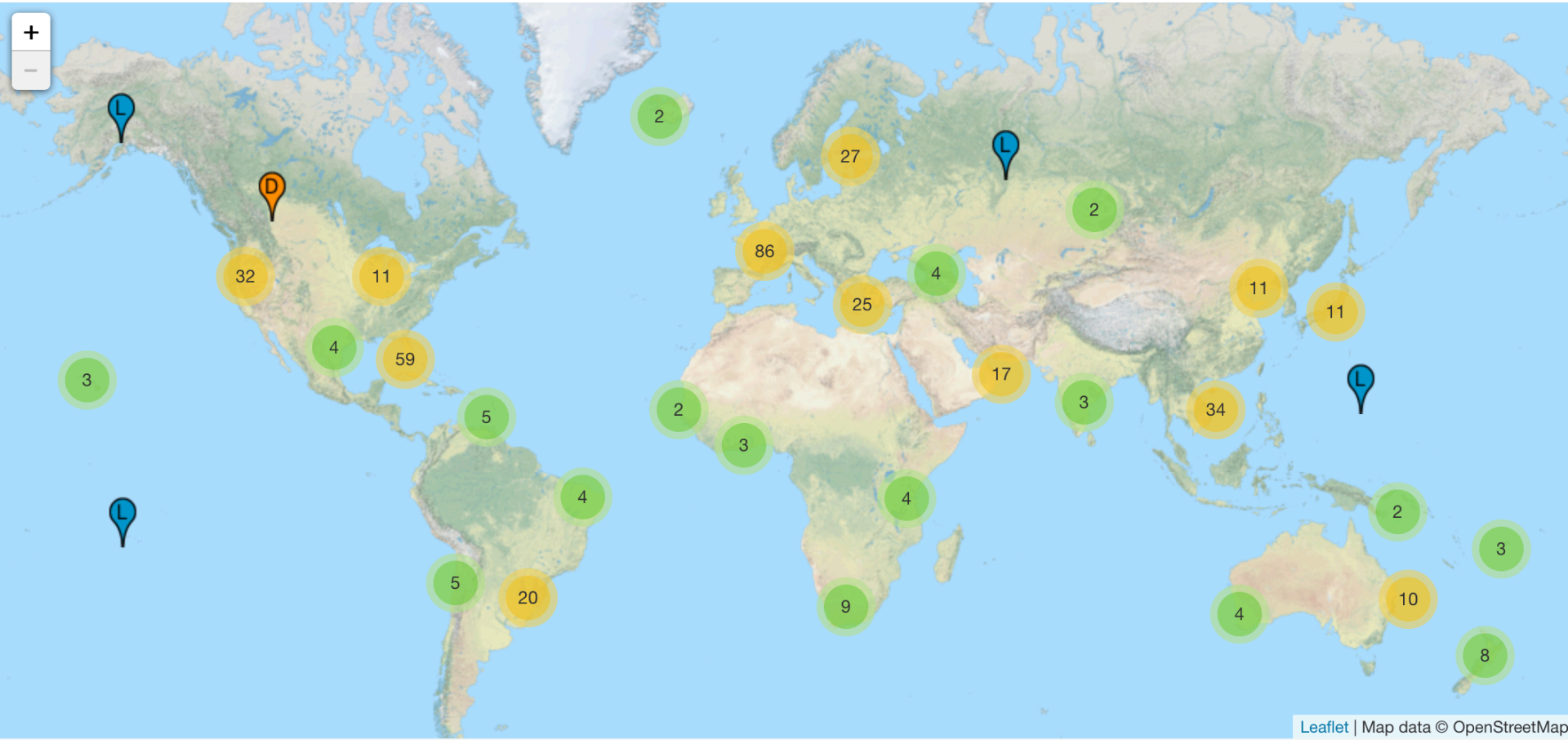
Observation: TLD servers typically cached in local name servers, thus root name servers not often visited

Robust DNS Service (Cont'ed)

- Redundant authoritative servers for each zone
  - "13" root servers --- are there just 13 boxes in the world?
- Caching resolvers learn DNS name to IP address mapping and *cache* the results locally
  - cache entries deleted after timeout (cache life time specified in the DNS query reply)
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited

DNS Root Name Servers

- 13 root name servers worldwide, operated by various parties on a coordinated, volunteering basis
  - All have multiple instances via anycast, see http://root-servers.org



http://www.iana.org/domains/root/servers

iana
Internet Assigned Numbers Authority

DOMAINS NUMBERS PROTOCOLS ABOUT IANA

Domain Names

- Overview
- **Root Zone Management**
  - Overview
  - Root Database
  - Hint and Zone Files
  - Change Requests
  - Instructions & Guides
  - **Root Servers**
- .INT Registry
- .ARPA Registry
- IDN Practices Repository
- Root Key Signing Key (DNSSEC)
- Reserved Domains

Root Servers

The authoritative name servers that serve the DNS root zone, commonly known as the "root servers", are a network of hundreds of servers in many countries around the world. They are configured in the DNS root zone as 13 named authorities, as follows.

List of Root Servers

Hostname	IP Addresses	Manager
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	192.228.79.201, 2001:500:84::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4	US Department of Defence (NIC)
h.root-servers.net	128.63.2.53, 2001:500:1::803f:235	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:3::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

DNS Protocol Details

DNS records

DNS: all DNS servers storing all data in *Resource Records* (RR)

RR format: **(name, TTL, class, type, data)**

Type=A (AAAA)

name is hostname

value is IPv4 (IPv6) address

Type=NS

- **name** is a domain (e.g. foo.com)
- **value** is hostname of authoritative name server for this domain

Type=CNAME

name is a alias name for some "canonical" (the real) name

`www.ibm.com` is really `servereast.backup2.ibm.com`

Type=MX

value is name of mailserver associated with **name**

e.g.

```
name = cs.ucla.edu
data= {10,
       mailman.cs.ucla.edu
type = MX
ttl = 172800
```

DNS Query/Reply

Same binary message format

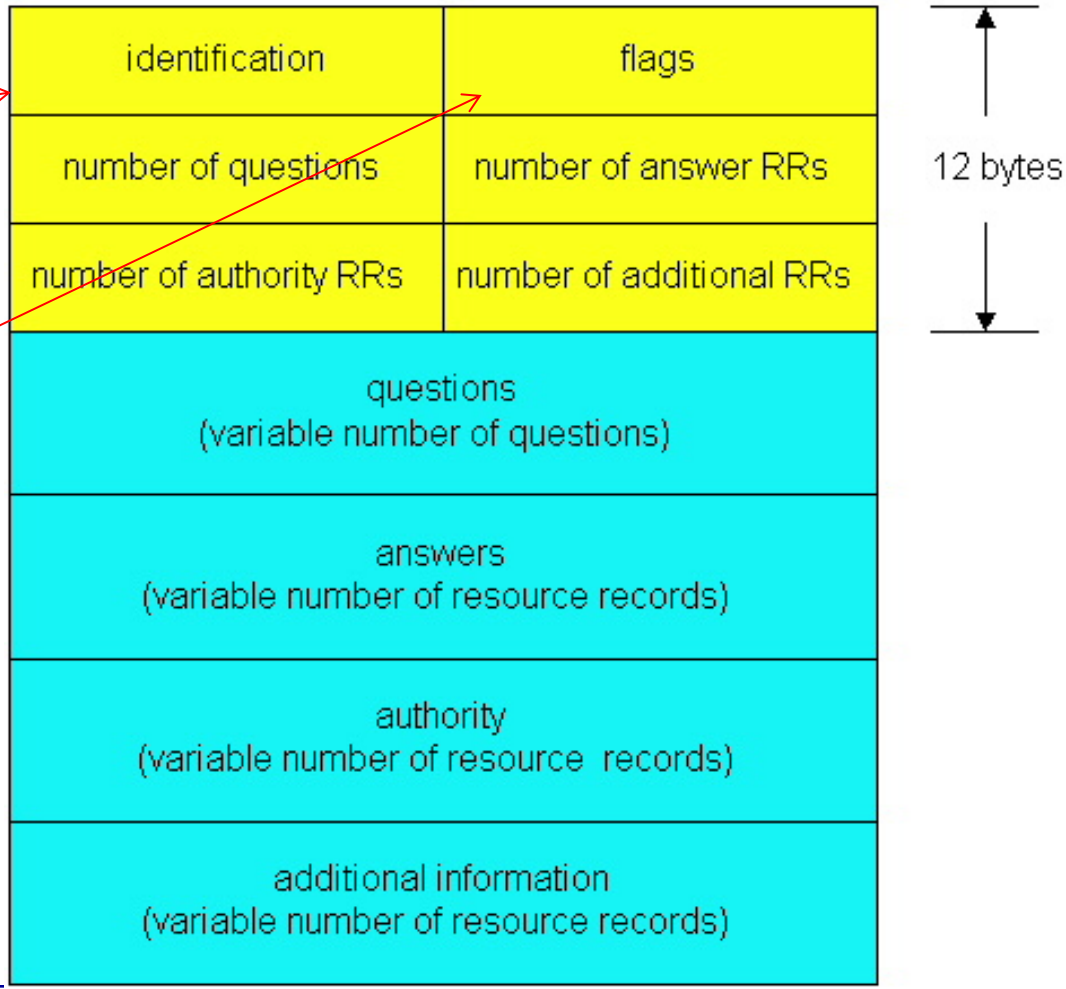
Message header

identification: 16-bit number

- reply to query uses same number

Flags:

- query or reply
- recursion desired
- recursion available
- reply is authoritative



Exploring DNS

- dig
  - Should be available by default on macOS
  - Part of “bind” package on Linux (and if brave enough, on Windows)
  - https://www.digwebinterface.com/

One example

```
lixia% dig berkeley.edu
. . . .
;; QUESTION SECTION:
;berkeley.edu.                  IN      A

;; ANSWER SECTION:
berkeley.edu.           300     IN      A       169.229.216.200

;; AUTHORITY SECTION:
berkeley.edu.           77441   IN      NS      adns1.berkeley.edu.
berkeley.edu.           77441   IN      NS      ns.v6.berkeley.edu.
berkeley.edu.           77441   IN      NS      phloem.uoregon.edu.
berkeley.edu.           77441   IN      NS      aodns1.berkeley.edu.
berkeley.edu.           77441   IN      NS      sns-pb.isc.org.
berkeley.edu.           77441   IN      NS      adns2.berkeley.edu.
berkeley.edu.           77441   IN      NS      aodns2.berkeley.edu.

;; ADDITIONAL SECTION:
ns.v6.berkeley.edu.     2052    IN      A       128.32.136.6
ns.v6.berkeley.edu.     2052    IN      AAAA    2607:f140:ffff:fffe::6
adns1.berkeley.edu.     83042   IN      A       128.32.136.3
adns1.berkeley.edu.     1117    IN      AAAA    2607:f140:ffff:fffe::3
adns2.berkeley.edu.     47835   IN      A       128.32.136.14
adns2.berkeley.edu.     1117    IN      AAAA    2607:f140:ffff:fffe::e
aodns1.berkeley.edu.    52369   IN      A       192.35.225.133
aodns1.berkeley.edu.    77441   IN      AAAA    2607:f010:3f8:8000::ff:fe00:53
aodns2.berkeley.edu.    147760  IN      A       128.253.35.148
phloem.uoregon.edu.     62440   IN      A       128.223.32.35
phloem.uoregon.edu.     148612  IN      AAAA    2001:468:d01:20::80df:2023
sns-pb.isc.org.         85841   IN      A       192.5.4.1
sns-pb.isc.org.         81969   IN      AAAA    2001:500:2e::1
```

```
✔  08:15 ~ $ dig cs.ucla.edu

; <<>> DiG 9.8.5-P1 <<>> cs.ucla.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13353
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 7, ADDITIONAL: 10

;; QUESTION SECTION:
;cs.ucla.edu.                   IN      A

;; ANSWER SECTION:
cs.ucla.edu.            14400   IN      A       164.67.100.172

;; AUTHORITY SECTION:
cs.ucla.edu.            14400   IN      NS      NS2.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS1.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS2.DNS.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS3.DNS.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS1.DNS.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS0.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS3.cs.ucla.edu.

;; ADDITIONAL SECTION:
NS0.cs.ucla.edu.  14400   IN      A       131.179.128.30
NS1.cs.ucla.edu.  14400   IN      A       131.179.128.16
NS1.DNS.ucla.edu. 7047    IN      A       192.35.225.7
NS1.DNS.ucla.edu. 56466   IN      AAAA    2607:f010:3fe:12::ff:fe01:35
NS2.cs.ucla.edu.  14400   IN      A       131.179.128.17
NS2.DNS.ucla.edu. 7047    IN      A       192.12.234.140
NS2.DNS.ucla.edu. 56466   IN      AAAA    2607:f140::e000:0:ff:fe01:35
NS3.cs.ucla.edu.  14400   IN      A       131.179.128.18
NS3.DNS.ucla.edu. 7047    IN      A       192.35.210.7
NS3.DNS.ucla.edu. 56466   IN      AAAA    2607:f600:8001:1::ff:fe01:35

;; Query time: 5 msec
;; SERVER: 131.179.128.30#53(131.179.128.30)
;; WHEN: Thu Jan 23 13:00:23 PST 2014
;; MSG SIZE  rcvd: 371
```

```
✔  08:15 ~ $ dig cs.ucla.edu mx
; <<>> DiG 9.8.5-P1 <<>> cs.ucla.edu mx
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24671
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 7, ADDITIONAL: 12

;; QUESTION SECTION:
;cs.ucla.edu.                   IN      MX

;; ANSWER SECTION:
cs.ucla.edu.            14400   IN      MX      3 Pelican.cs.ucla.edu.
cs.ucla.edu.            14400   IN      MX      13 Mailman.cs.ucla.edu.

;; AUTHORITY SECTION:
cs.ucla.edu.            14400   IN      NS      NS2.DNS.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS0.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS1.DNS.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS2.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS3.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS1.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS3.DNS.ucla.edu.

;; ADDITIONAL SECTION:
Pelican.cs.ucla.edu.    14400   IN      A       131.179.128.17
Mailman.cs.ucla.edu.    14400   IN      A       131.179.128.30
NS0.cs.ucla.edu.        14400   IN      A       131.179.128.30
NS1.cs.ucla.edu.        14400   IN      A       131.179.128.16
NS1.DNS.ucla.edu.       5862    IN      A       192.35.225.7
NS1.DNS.ucla.edu.       55281   IN      AAAA    2607:f010:3fe:12::ff:fe01:35
NS2.cs.ucla.edu.        14400   IN      A       131.179.128.17
NS2.DNS.ucla.edu.       5862    IN      A       192.12.234.140
NS2.DNS.ucla.edu.       55281   IN      AAAA    2607:f140::e000:0:ff:fe01:35
NS3.cs.ucla.edu.        14400   IN      A       131.179.128.18
NS3.DNS.ucla.edu.       5862    IN      A       192.35.210.7
NS3.DNS.ucla.edu.       55281   IN      AAAA    2607:f600:8001:1::ff:fe01:35

;; Query time: 4 msec
;; SERVER: 131.179.128.30#53(131.179.128.30)
;; WHEN: Thu Jan 23 13:20:09 PST 2014
;; MSG SIZE  rcvd: 435
```

```
✔  08:15 ~ $ dig cs.ucla.edu ns

; <<>> DiG 9.8.5-P1 <<>> cs.ucla.edu ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58184
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 10

;; QUESTION SECTION:
;cs.ucla.edu.                   IN      NS

;; ANSWER SECTION:
cs.ucla.edu.            14400   IN      NS      NS0.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS2.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS3.DNS.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS3.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS2.DNS.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS1.cs.ucla.edu.
cs.ucla.edu.            14400   IN      NS      NS1.DNS.ucla.edu.

;; ADDITIONAL SECTION:
NS0.cs.ucla.edu.    14400   IN      A       131.179.128.30
NS1.cs.ucla.edu.    14400   IN      A       131.179.128.16
NS1.DNS.ucla.edu.   7149    IN      A       192.35.225.7
NS1.DNS.ucla.edu.   56568   IN      AAAA    2607:f010:3fe:12::ff:fe01:35
NS2.cs.ucla.edu.    14400   IN      A       131.179.128.17
NS2.DNS.ucla.edu.   7149    IN      A       192.12.234.140
NS2.DNS.ucla.edu.   56568   IN      AAAA    2607:f140::e000:0:ff:fe01:35
NS3.cs.ucla.edu.    14400   IN      A       131.179.128.18
NS3.DNS.ucla.edu.   7149    IN      A       192.35.210.7
NS3.DNS.ucla.edu.   56568   IN      AAAA    2607:f600:8001:1::ff:fe01:35

;; Query time: 11 msec
;; SERVER: 131.179.128.30#53(131.179.128.30)
;; WHEN: Thu Jan 23 12:58:41 PST 2014
;; MSG SIZE  rcvd: 355
```

Searching the Truth (1/2)

```
✔  22:39 ~ $ dig +noall +answer amazon.com a @8.8.8.8
amazon.com.             39      IN      A       54.239.17.7
amazon.com.             39      IN      A       54.239.25.200
amazon.com.             39      IN      A       54.239.26.128
amazon.com.             39      IN      A       54.239.25.208
amazon.com.             39      IN      A       54.239.17.6
amazon.com.             39      IN      A       54.239.25.192
```

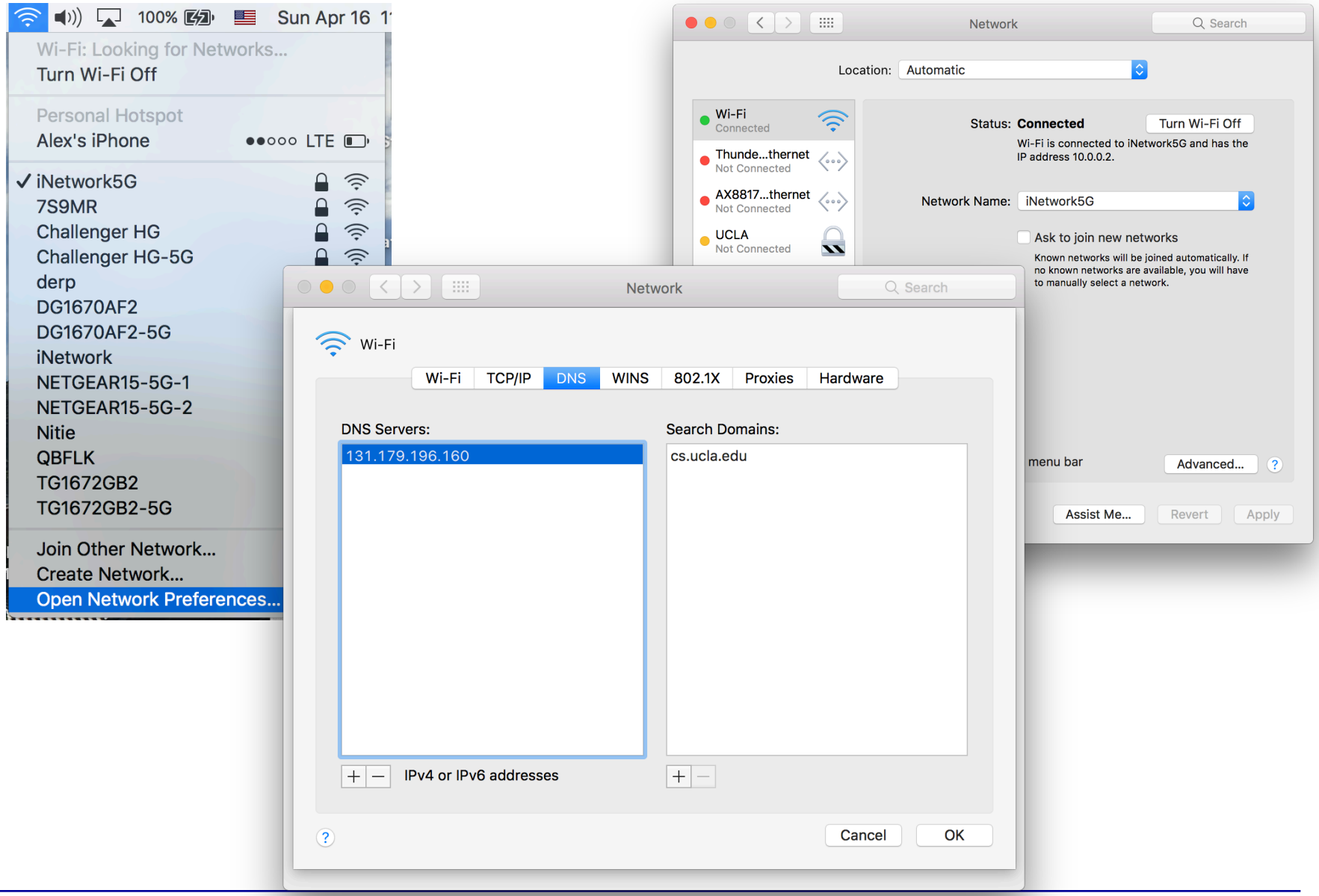
```
✔  22:35 ~ $ dig +noall +answer amazon.com a @131.179.196.160
amazon.com.             3600    IN      A       54.239.26.128
amazon.com.             3600    IN      A       54.239.17.7
amazon.com.             3600    IN      A       54.239.17.6
amazon.com.             3600    IN      A       54.239.25.208
amazon.com.             3600    IN      A       54.239.25.192
amazon.com.             3600    IN      A       54.239.25.200
```

Searching the Truth (2/2)

```
✔ 22:35 ~ $ dig +noall +answer www.amazon.com a @8.8.8.8
www.amazon.com.           1461    IN      CNAME   www.cdn.amazon.com.
www.cdn.amazon.com.       238     IN      CNAME
        d3ag4hukkh62yn.cloudfront.net.
d3ag4hukkh62yn.cloudfront.net. 59 IN      A       54.230.140.143
d3ag4hukkh62yn.cloudfront.net. 59 IN      A       54.230.140.28
d3ag4hukkh62yn.cloudfront.net. 59 IN      A       54.230.140.212
d3ag4hukkh62yn.cloudfront.net. 59 IN      A       54.230.140.148
```

```
✔  22:35 ~ $ dig +noall +answer www.amazon.com a @131.179.196.160
www.amazon.com.           3600    IN      A       131.179.196.70
```

If you want to (temporarily) set different caching DNS resolver



Reading Assignment

- **An Illustrated Guide to the Kaminsky DNS Vulnerability**
  - http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

Challenge for This Week

- DNS Scavenger hunt
  - Find as many IPv4 addresses for google.com as you can (real ones)
    - Minimum for submission: > 20
    - Will be updating top 5 on piazza

Other Questions

- Caching of responses is one two components that makes DNS a robust service
  - (remember what is/are the other(s))?

- What if one sends DNS queries for records to don't exist?
  - Does DNS still scale?

Inserting records into DNS

- Example: assume creating a new "Foo University"
- Register name foo.edu at EDU registrar
  - Need to provide registrar with names and IP addresses of your authoritative name servers (primary and secondary)
  - Registrar inserts two RRs into the edu TLD server:

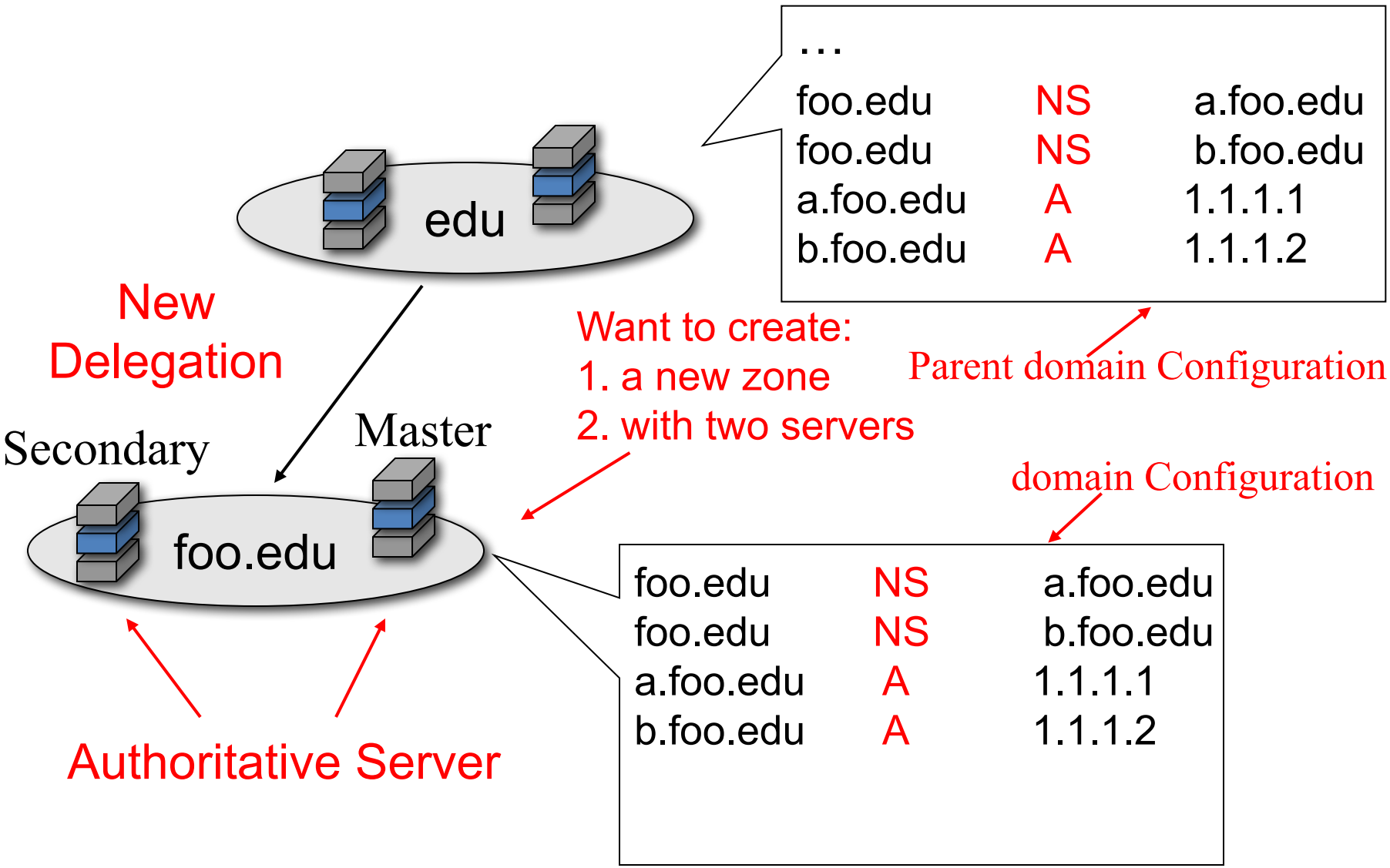
```
(foo.edu, a.foo.edu, NS)
(a.foo.edu, 1.1.1.1, A)
```

- Put in authoritative server Type A record for www.foo.edu, and Type MX record for foo.edu

How do people get the IP address of Web site `www.foo.edu`?

FYI

Example Configuration



Food for thought: why not a centralize DNS?

- single point of failure
- traffic volume
- Maintenance (keeping all RRs updated)
- Long distance between database servers and clients
- *The most important factor: the need for distributed management*