College of Information Studies

University of Maryland Hornbake Library Building College Park, MD 20742-4345

Domain Names

Session 6

INST 346

Technologies, Infrastructure and Architecture

Quiz

- Start: 5:00 sharp
- End: 5:05 sharp (pencil down or zero credit)
- Include your name!
- No communication with anyone till 5:05
  - No email, no talking, no SMS, no chat, …
  - Even if you finish early!
- Open book, open notes, open Web, open mind

Muddiest Points

- Socket programming
  - How much programming is in this class?
- Difference between POP3 and IMAP
- Blocking on socket input

Goals for Today

- Demo socket programming
- Domain Name System
- Preview H2

Socket Programming Demo

DNS: domain name system

people: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address (32 bit) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

Q: how to map between IP address and name, and vice versa ?

Domain Name System:

- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

Designing DNS

Centralized design – problems :

1. Single point of failure
2. Traffic volume
3. Distant centralized database
4. Maintenance

DNS

A distributed database

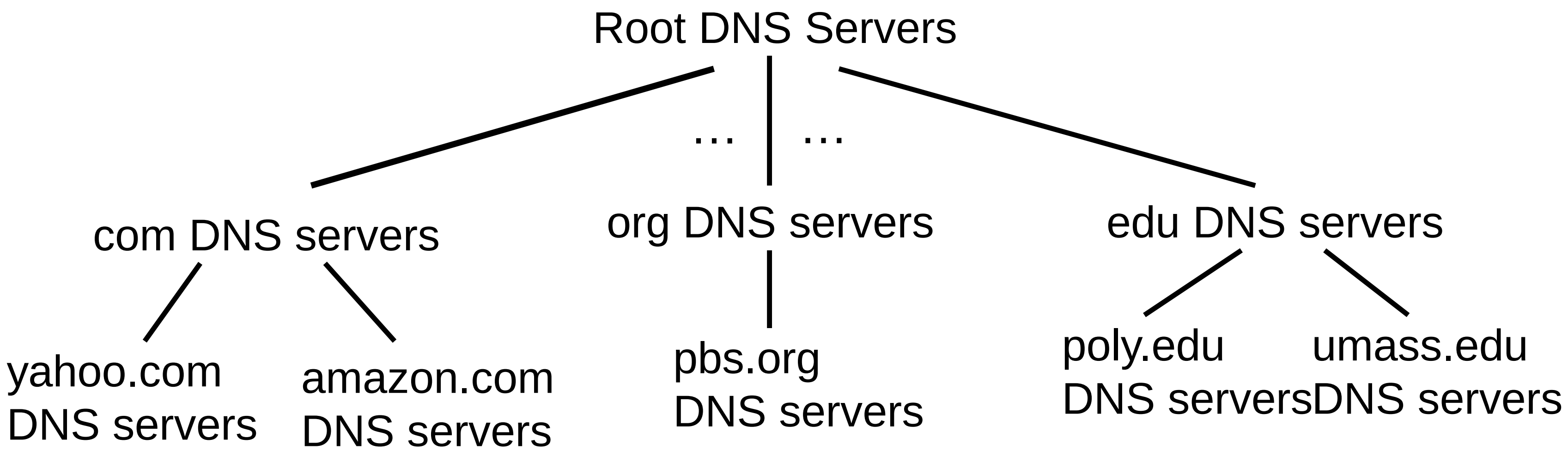
Implemented on a hierarchy of DNS servers

An application-layer protocol that allows hosts to query the distributed database.

Runs over UDP, Port 53

Employed by other app level protocols

DNS: a distributed, hierarchical database

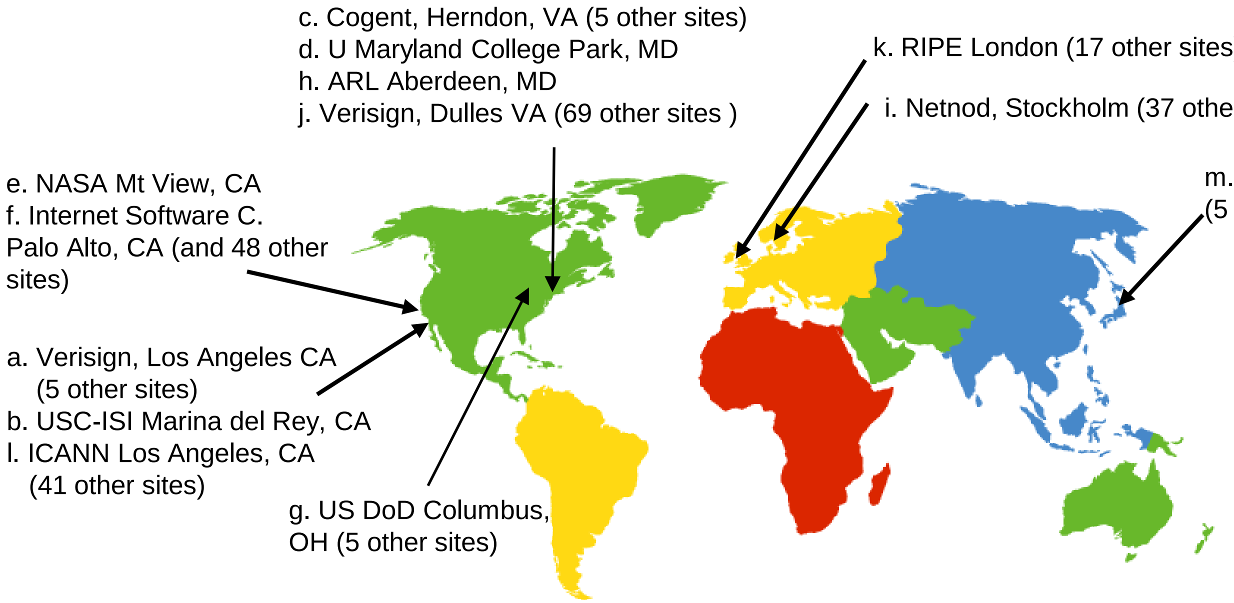


client wants IP for www.amazon.com; 1st approximation:

- client queries root server to find com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: root name servers

- contacted by local name server that can not resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server



c. Cogent, Herndon, VA (5 other sites)
d. U Maryland College Park, MD
h. ARL Aberdeen, MD
j. Verisign, Dulles VA (69 other sites)

k. RIPE London (17 other sites)

i. Netnod, Stockholm (37 other sites)

e. NASA Mt View, CA
f. Internet Software C.
Palo Alto, CA (and 48 other sites)

m. WIDE Tokyo
(5 other sites)

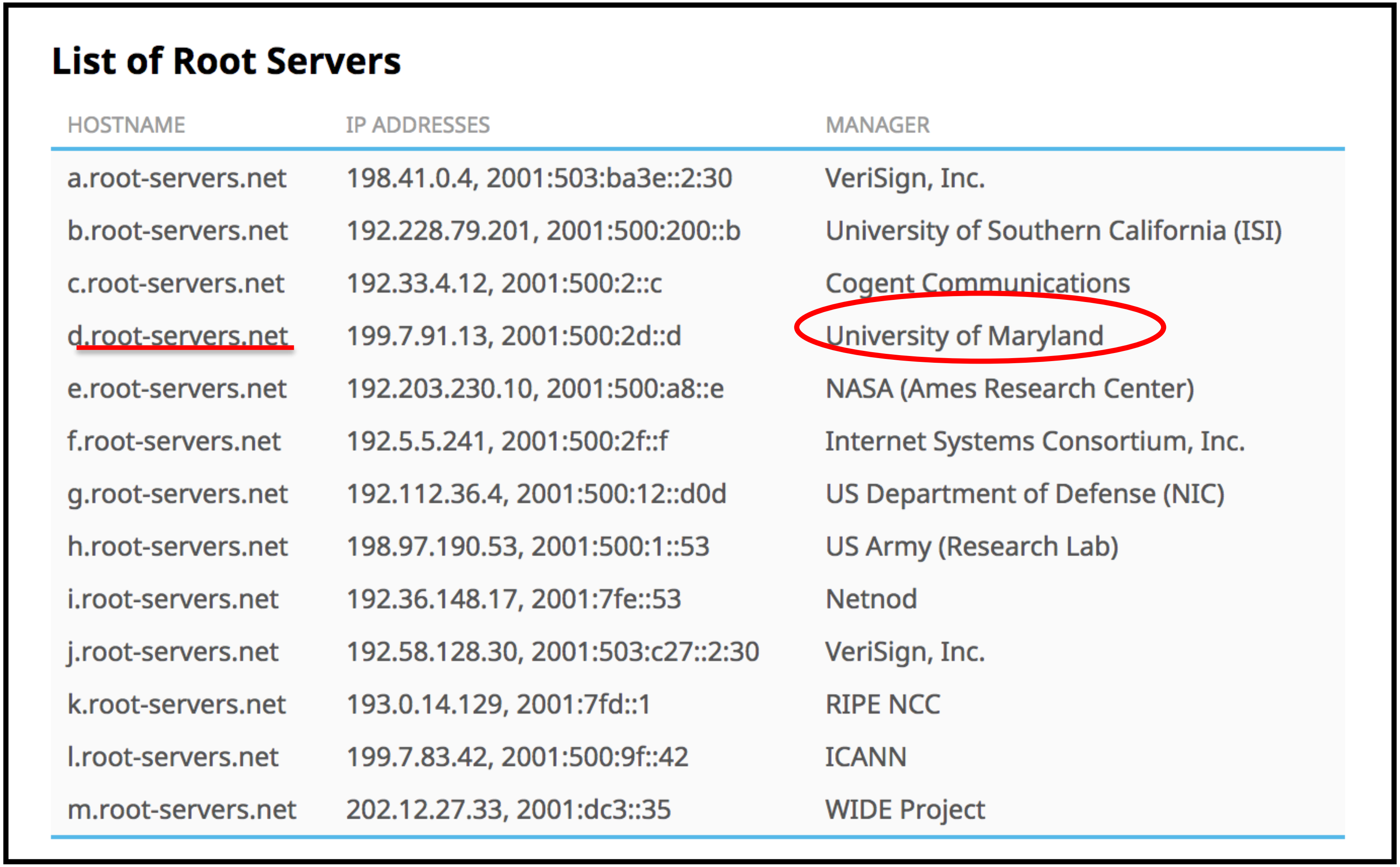
a. Verisign, Los Angeles CA
(5 other sites)
b. USC-ISI Marina del Rey, CA
l. ICANN Los Angeles, CA
(41 other sites)

g. US DoD Columbus, OH (5 other sites)

13 logical root name "servers" worldwide

- *each "server" replicated many times*

13 Root Name Servers



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, 2001:500:200::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, 2001:500:a8::e	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, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	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, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

Source: https://www.iana.org/domains/root/servers

Local DNS name server

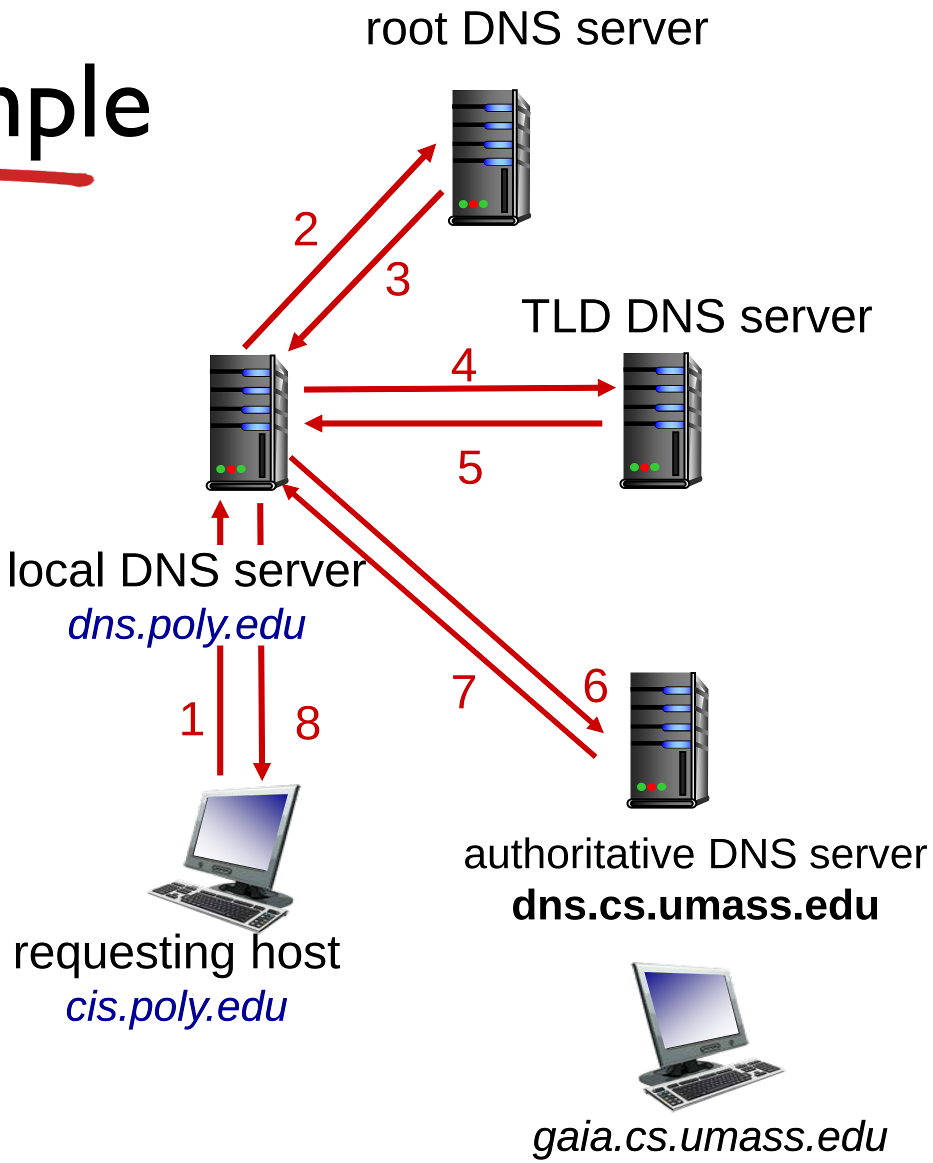
- does not strictly belong to hierarchy
- each ISP (residential ISP, company, university) has one
  - also called “default name server”
- when host makes DNS query, query is sent to its local DNS server
  - has local cache of recent name-to-address translation pairs (but may be out of date!)
  - acts as proxy, forwards query into hierarchy

DNS name resolution example

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu

iterated query:

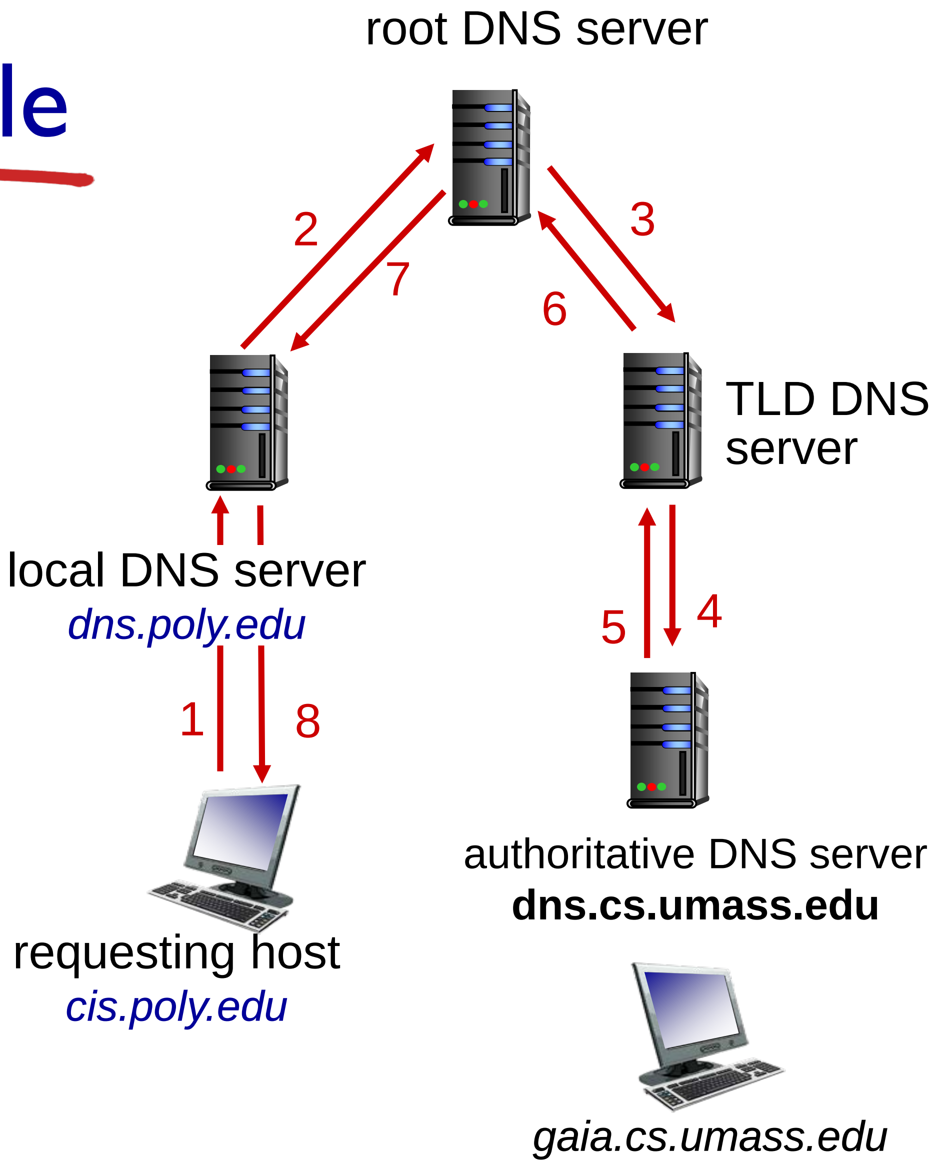
- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS name resolution example

recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



DNS: caching, updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time (TTL)
  - TLD servers typically cached in local name servers
    - thus root name servers not often visited
- cached entries may be *out-of-date* (best effort name-to-address translation!)
  - if name host changes IP address, may not be known Internet-wide until all TTLs expire

DNS records

DNS: distributed database storing resource records (RR)

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

type=A

- **`name`** is hostname
- **`value`** is IP address

type=NS

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

type=CNAME

- **`name`** is alias name for some "canonical" (the real) name
- **`www.ibm.com`** is really **`servereast.backup2.ibm.com`**
- **`value`** is canonical name

type=MX

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

DNS Aliasing

- Complicated canonical names, better hit rate
- Host name aliasing
- Mail server aliasing

Example:

1. All redirect to facebook.com ←

facebook.org; facebook.us; facebook.in

2. gmail.com → mail.google.com

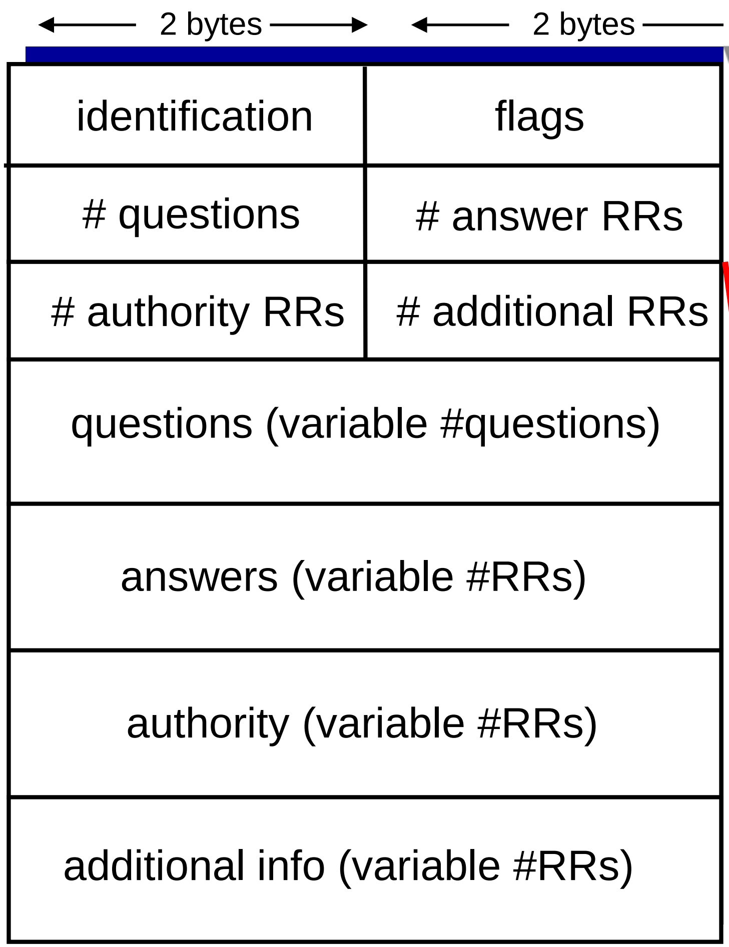
Example DNS lookup

Try on your command line or terminal :

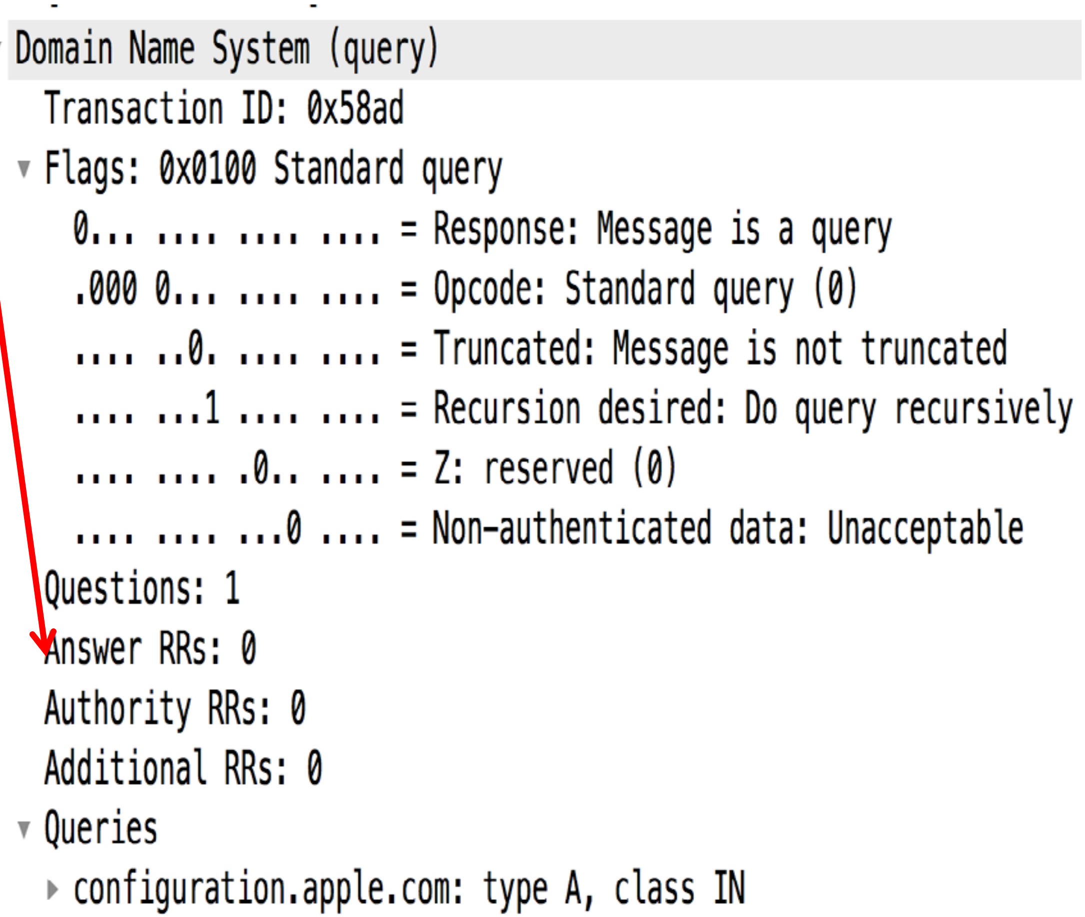
```
$ nslookup www.umd.edu
```

DNS protocol, messages

- *query* and *reply* - both same *message format*

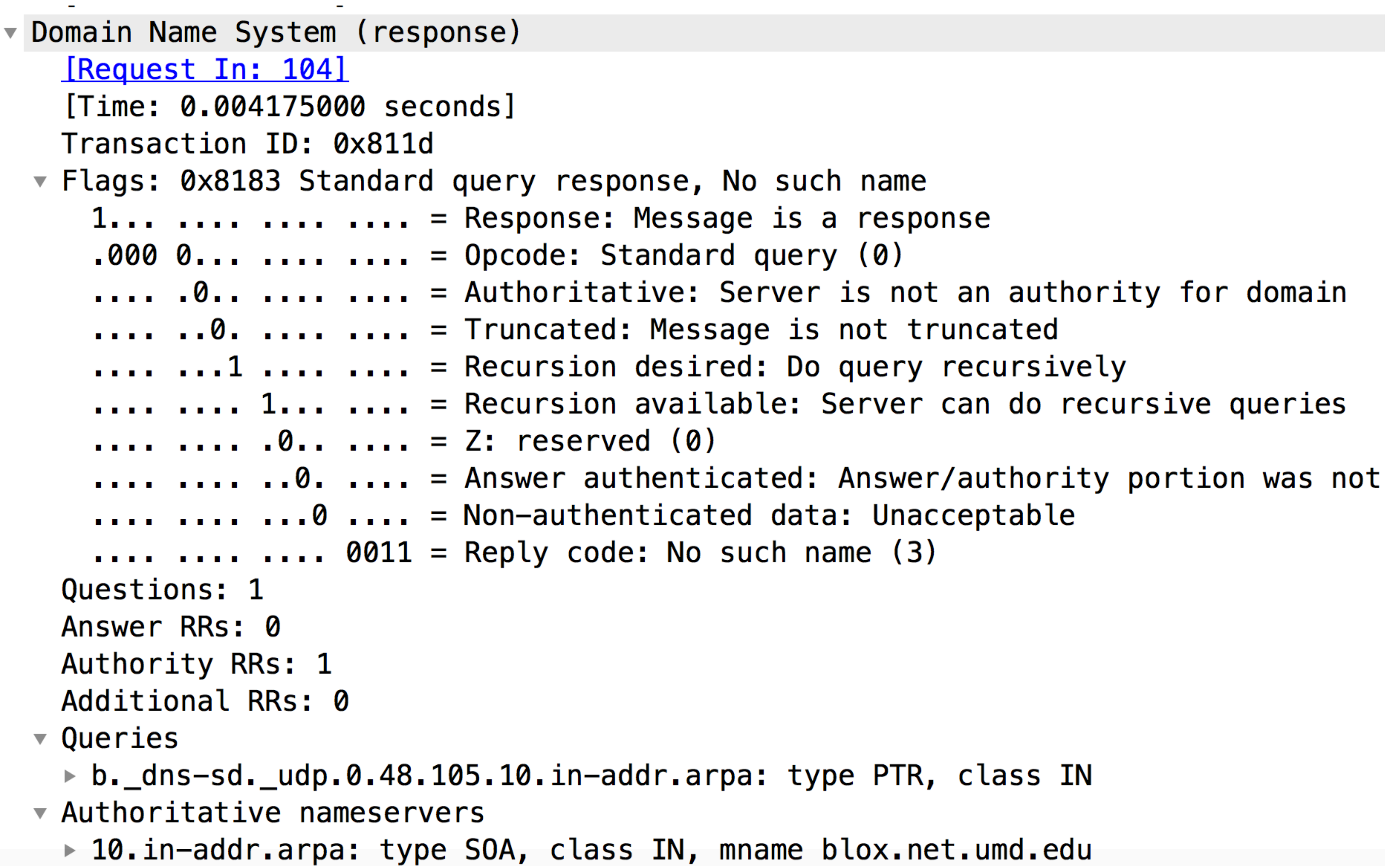


2 bytes	2 bytes
identification	flags
# questions	# answer RRs
# authority RRs	# additional RRs
questions (variable #questions)	
answers (variable #RRs)	
authority (variable #RRs)	
additional info (variable #RRs)	



```
▼ Domain Name System (query)
    Transaction ID: 0x58ad
  ▼ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▸ configuration.apple.com: type A, class IN
```

DNS Reply message



```
▼ Domain Name System (response)
    [Request In: 104]
    [Time: 0.004175000 seconds]
    Transaction ID: 0x811d
  ▼ Flags: 0x8183 Standard query response, No such name
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0011 = Reply code: No such name (3)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  ▼ Queries
    ▶ b._dns-sd._udp.0.48.105.10.in-addr.arpa: type PTR, class IN
  ▼ Authoritative nameservers
    ▶ 10.in-addr.arpa: type SOA, class IN, mname blox.net.umd.edu
```

Inserting records into DNS

- example: new startup "Network Utopia"
- register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into .com TLD server:

    ```
    (networkutopia.com, dns1.networkutopia.com, NS)
    (dns1.networkutopia.com, 212.212.212.1, A)
    ```
- create authoritative server type A record for www.networkuptopia.com; type MX record for networkutopia.com

DNS takeaways

- Translating domain names to IP addresses
- Distributed, hierarchical database design
- Root, TLD, authoritative, local DNS servers
- Resource Records (RR)
- Aliasing
- Caching, Updating, Inserting RRs
- DNS protocol and messages

H2 Preview

Before You Go

On a sheet of paper, answer the following (ungraded) question (no names, please):

Name one thing the instructor could do to improve her teaching