Lecture 6B

DNS and HTTP

DNS: Domain Name System

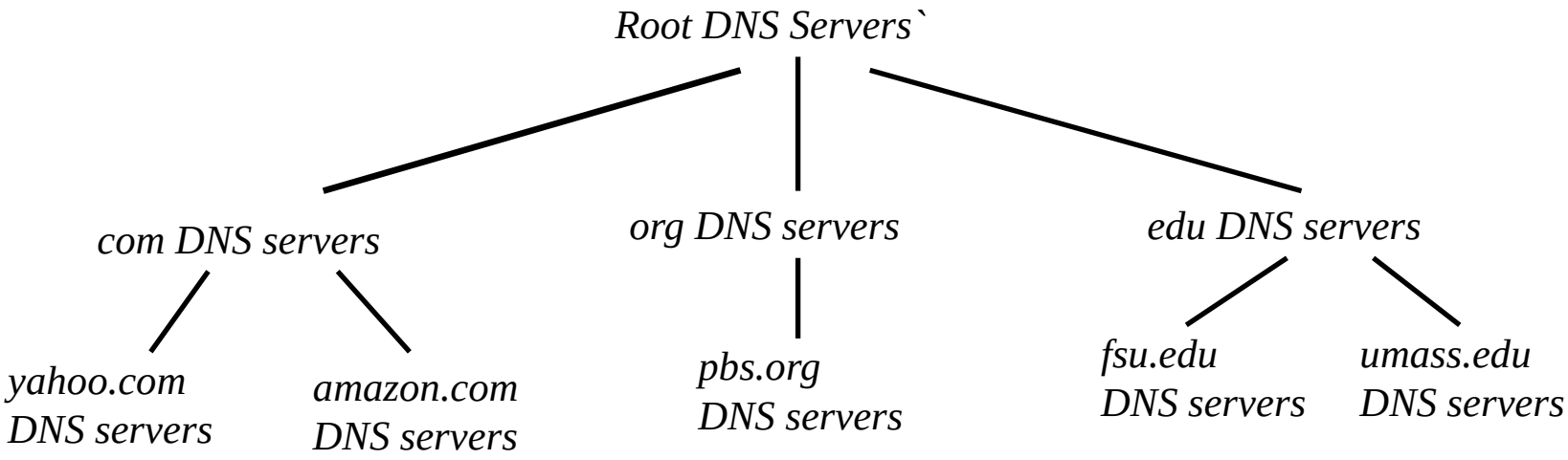
DNS services

- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

Features

- Scalable – not a single file of name / address mappings
- Hierarchical – reduces host name conflicts
- Distributed

Distributed, Hierarchical Database



- Below ROOT, we have Top-Level Domain (TLD). Ex: In www.example.com, the TLD is .com.

- The next level of domain hierarchy is second-level domain which are usually assigned to specific entities such as companies, schools etc.

- The domain namespace is organized in a hierarchical tree-like structure.
- Each node is called a domain, or subdomain.
- The root of the domain is called ROOT, denoted as ‘ . ‘.

Internet Agencies

- **IETF** Internet Engineering Task Force *open standards, non profit*

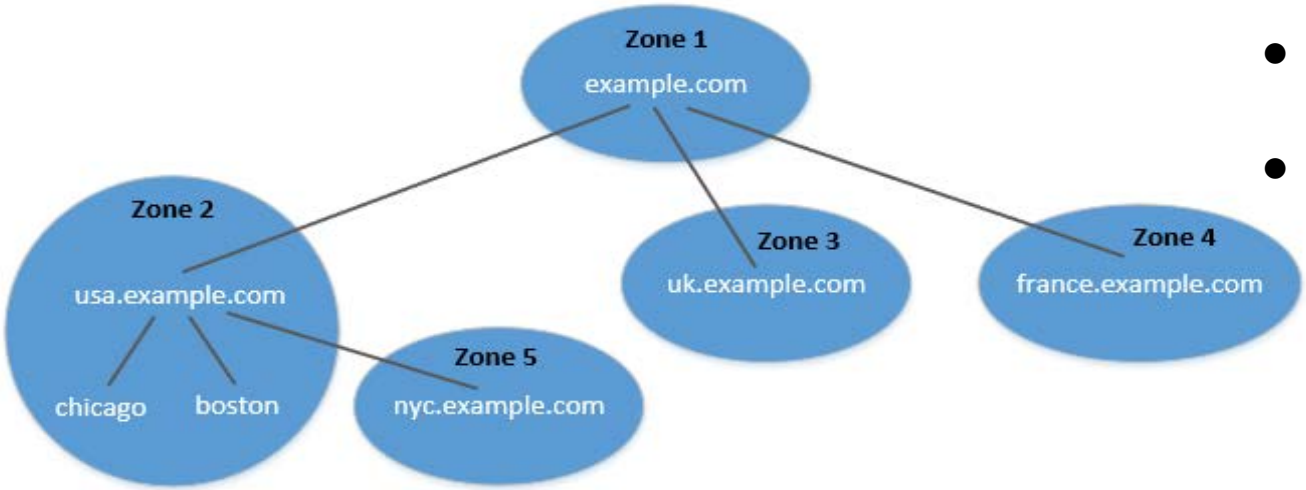
- **IANA** Internet Assigned Numbers Authority *domain services (maintains DNS database, is managed by ICANN)*

- **ICANN** Internet Corporation for Assigned Names and Numbers *Independent non-profit organization; members include governments, companies, etc.*

Top Level Domain (TLD)

- Infrastructure TLD: .arpa *Restricted to IANA on behalf of IETF*
- Generic TLD (gTLD): .com, .info, .net, .org, (*3 or more characters, generic*)
- Sponsored TLD (sTLD): .edu, .gov, .mil, .travel, .jobs (*sponsored by private agencies or organizations that establish and enforce rules restricting eligibility to use the TLD*)
- Country Code TLD (ccTLD): .au (Australia), .cn (China), .fr (France)
- Reserved TLD: .example, .test, .localhost, .invalid
- There are many more than a thousand TLDs

DNS Zone



- DNS is organized according to zones.
- A zone groups contiguous domains and subdomains on the domain tree and assign management authority to an entity.

- The tree structure depicts subdomains within example.com domain.
- In this case, there are multiple DNS zones one for each country. The zone keeps records of who the authority is for each of its subdomains.
- The zone for example.com contains only the DNS records for the hostnames that do not belong to any subdomain like mail.example.com

Zone vs Domain

- A DNS zone only contains a portion of the DNS data for a domain.
- If a domain is not divided into subdomains, the zone and domain are essentially the same, because the zone contains all the DNS data for the domain.
- When a domain is divided into subdomains, their DNS data can still be put in the same zone, so domain and zone are still the same.
- But subdomains can have their own zones.
- usa.example.com is a domain with subdomains as boston, nyc and chicago. Two zones are created for usa.example.com. First contains usa domain, chicago and boston subdomain and second contains nyc subdomain.

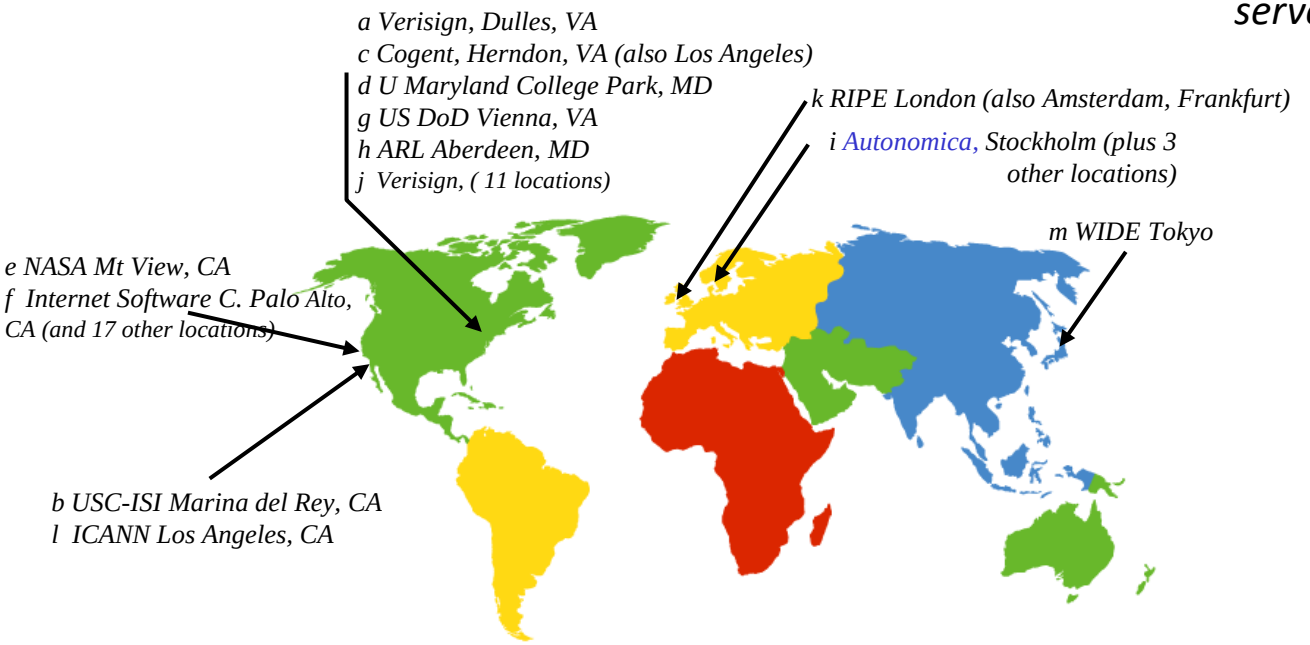
Authoritative Name Servers

- Each DNS zone has at least one authoritative nameserver that publishes information about the zone.
- It provides the original and definitive answers to DNS queries.
- An authoritative name server can be a master server (primary) or slave server (secondary).
- A master server stores the master copies of all zone records whereas a slave server uses an automatic updating mechanism to maintain an identical copy of the master records.

DNS: Root name servers

- contacted by local name server that cannot resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server
  - http://root-servers.org/

13 authoritative root name servers worldwide



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

13 DNS Root 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	199.9.14.201, 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

Local Name Server

- Each ISP (residential ISP, company, university) has one.
  - Also called “default name server”
- When a host makes a DNS query, query is sent to its local DNS server
  - Acts as a proxy, forwards query into hierarchy.

Anycasting, Unicasting, Multicasting, Broadcasting

- *Unicasting* 1 to 1: standard routing of a packet given source and destination IP

- *Broadcasting* 1 to all: used in a local net, uses a broadcast address Ethernet and IPv4 level (note special values are used such as all 1's or 1's in "host" positions

- *Multicasting* 1 to many: all receivers will get the packet. Special name space in IPv4

- *Anycasting* 1 to (1 of many): many machines have the same IP address but only one receives, typically physically closest machine. Note this requires special actions of routers and is supported through BGP advertisements (packets follow shortest path)

CloudFlare

- Cloudflare, Inc. is an American web-infrastructure and website-security company, providing content-delivery-network services, DDoS mitigation, Internet security, and distributed domain-name-server services.
- Data Centers (PoPs) in 335 cities, 125+ countries)
- One of the IPs that CloudFlare might announce for DNS services is 173.245.58.205
- A route to that IP address is announced from a number of CloudFlare data centers, say 25.
- Those routers look at the available paths to CloudFlare's end points and send the packet down the one with the fewest stops along the way

Anycasting example

$traceroute 173.245.58.205 CloudFlare SF

traceroute to 173.245.58.205 (173.245.58.205), 64 hops max, 52 byte packets

1 192.168.2.1 (192.168.2.1) 3.473 ms 1.399 ms 1.247 ms

2 10.10.11.1 (10.10.11.1) 3.136 ms 2.857 ms 3.206 ms

3 ge-0-2-5.cr1.sfo1.us.nlayer.net (69.22.X.X) 2.936 ms 3.405 ms 3.193 ms

4 ae3-70g.cr1.pao1.us.nlayer.net (69.22.143.170) 3.638 ms 4.076 ms 3.911 ms

5 ae1-70g.cr1.sjc1.us.nlayer.net (69.22.143.165) 4.833 ms 4.874 ms 4.973 ms

6 ae1-40g.ar2.sjc1.us.nlayer.net (69.22.143.118) 8.926 ms 8.529 ms 6.742 ms

7 as13335.xe-8-0-5.ar2.sjc1.us.nlayer.net (69.22.130.146) 5.048 ms

8 173.245.58.205 (173.245.58.205) 4.601 ms 4.338 ms 4.611 ms

Anycasting example

$traceroute 173.245.58.205 CloudFlare London

traceroute to 173.245.58.205 (173.245.58.205), 30 hops max, 60 byte packets

1 212.111.X.X (212.111.X.X) 6.574 ms 6.514 ms 6.522 ms

2 212.111.33.X (212.111.33.X) 0.934 ms 0.935 ms 0.969 ms

3 85.90.238.69 (85.90.238.69) 1.396 ms 1.381 ms 1.405 ms

4 ldn-b3-link.telia.net (80.239.167.93) 0.700 ms 0.696 ms 0.670 ms

5 ldn-bb1-link.telia.net (80.91.247.24) 2.349 ms 0.700 ms 0.671 ms

6 ldn-b5-link.telia.net (80.91.246.147) 0.759 ms 0.771 ms 0.774 ms

7 cloudflare-ic-154357-ldn-b5.c.telia.net (80.239.161.246) 0.917 ms 0.853 ms 0.833 ms

8 173.245.58.205 (173.245.58.205) 0.972 ms 1.292 ms 0.916 ms

Local DNS Files

- **`/etc/host`**: stores IP addresses for some hostnames. Before machine contacts the local DNS servers, it first looks into this file for the IP address.

```
127.0.0.1    localhost
127.0.0.1    www.CSRFLabAttacker.com
127.0.0.1    www.CSRFLabElgg.com
127.0.0.1    www.XSSLabElgg.com
```

- **`/etc/resolv.conf`**: provide information to the machine's DNS resolver about the IP address of the local DNS server. The IP address of the local DNS server provided by DHCP is also stored here.

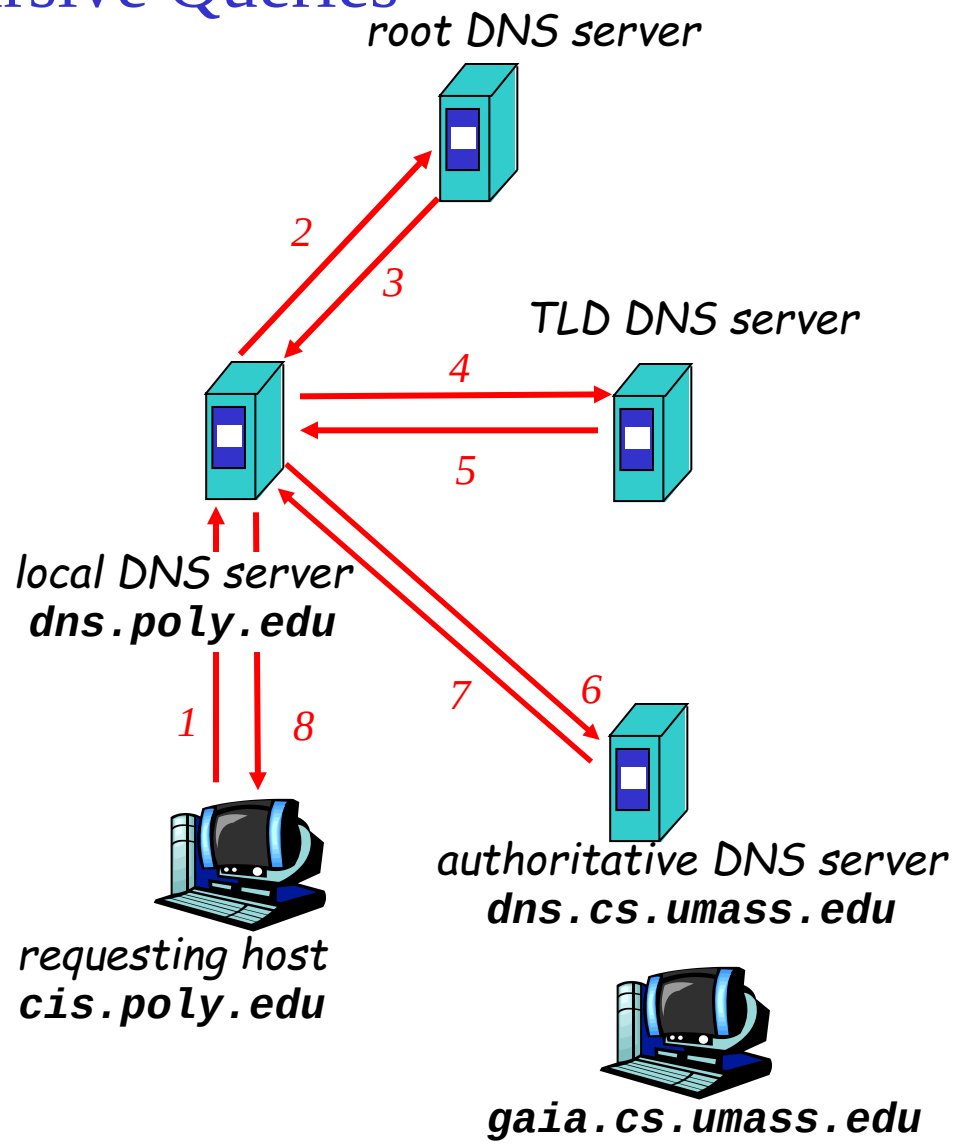
Iterative Queries &Recursive Queries

<u>iterated query:</u>

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

<u>recursive query:</u>

- contacted server is responsible for getting the information back to me
- <u>Example:</u>
  - cis.poly.edu wants the IP address of gaia.cs.umass.edu
  - the local DNS server is configured for recursive queries
  - Step 1 is a recursive query to the local server coming from the resolver component of host and the result is step 8
  - All other queries are iterative
  - root, TLD, and authoritative are *not* configured to handle recursive queries



DNS: caching and updating records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited

DNS records

DNS: distributed db storing resource records (RR)

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

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

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

- *Type=MX*
  - ***value*** *is name of mail server associated with the* ***name*** *domain*

More on record information

- class = IN: this refers to TCP/IP records (internet)
- type = MX: mail exchange record
  - What are the mail servers for this domain
- type = SOA: start of authority record
  - Includes primary name server, responsible party for the domain
- type = A: address record
  - Links domain name to IP address
- type = NS: name server record
  - Authoritative DNS servers for the domain; usually have a primary and secondary
- type = CN: canonical name record
  - Give the actual name (canonical name) of server given the well-known name

Tools to get DNS Information

- dig
- whois
- ping
- traceroute
- Try using this site:

  http://www.centralops.net

  - www.google.com
  - www.fsu.edu
  - cs.fsu.edu
  - 128.186.120.1

The dig command

```
[root@dragonwell ~]# dig www.google.com

; <<>> DiG 9.3.3rc2 <<>> www.google.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62363
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         56132   IN      CNAME   www.l.google.com.
www.l.google.com.       276     IN      A       74.125.157.147
www.l.google.com.       276     IN      A       74.125.157.99
www.l.google.com.       276     IN      A       74.125.157.103
www.l.google.com.       276     IN      A       74.125.157.104
www.l.google.com.       276     IN      A       74.125.157.105
www.l.google.com.       276     IN      A       74.125.157.106

;; Query time: 2 msec
;; SERVER: 128.186.120.179#53(128.186.120.179)
;; WHEN: Sat Apr 17 23:35:23 2010
;; MSG SIZE  rcvd: 148
```

The dig command

```
[root@dragonwell ~]# dig mail.ece.sunysb.edu

; <<>> DiG 9.3.3rc2 <<>> mail.ece.sunysb.edu
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 119
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;mail.ece.sunysb.edu.           IN      A

;; ANSWER SECTION:
mail.ece.sunysb.edu.    3600    IN      A       129.49.69.99

;; AUTHORITY SECTION:
ece.sunysb.edu.         3600    IN      NS      lab04.ee.sunysb.edu.
ece.sunysb.edu.         3600    IN      NS      CMENOC.sunysb.edu.
ece.sunysb.edu.         3600    IN      NS      eegw.ece.sunysb.edu.

;; ADDITIONAL SECTION:
eegw.ece.sunysb.edu.    3600    IN      A       129.49.68.100
lab04.ee.sunysb.edu.    3600    IN      A       129.49.68.184
CMENOC.sunysb.edu.      1800    IN      A       129.49.7.3

;; Query time: 65 msec
;; SERVER: 128.186.120.179#53(128.186.120.179)
;; WHEN: Sat Apr 17 23:34:23 2010
;; MSG SIZE  rcvd: 164
```

Emulating Local DNS Server (Step 1: Ask ROOT)

Directly send the query to this server.

```
seed@ubuntu:~$ dig @a.root-servers.net www.example.net

(Only a portion of the reply is shown here)
;; QUESTION SECTION:
;www.example.net.               IN      A

;; AUTHORITY SECTION:
net.                    172800  IN      NS      m.gtld-servers.net.
net.                    172800  IN      NS      l.gtld-servers.net.
net.                    172800  IN      NS      k.gtld-servers.net.

;; ADDITIONAL SECTION:
m.gtld-servers.net.     172800  IN      A       192.55.83.30
l.gtld-servers.net.     172800  IN      A       192.41.162.30
k.gtld-servers.net.     172800  IN      A       192.52.178.30
```

No answer (the root does not know the answer)

Go ask them!

DNS Response

There are 4 types of sections in a DNS response :

- Question section : Describes a question to a nameserver
- Answer section : Records that answer the question
- Authority section : Records that point toward authoritative nameservers
- Additional section : Records that are related to the query.

In the above example, we see that as root server doesn't know the answer there is no answer section, but tells us about the authoritative nameservers (NS Record) along with their IP addresses in the Additional section (A record).

Steps 2-3: Ask .net & example.net servers

```
seed@ubuntu:~$ dig @m.gtld-servers.net www.example.net

;; QUESTION SECTION:
;www.example.net.               IN      A

;; AUTHORITY SECTION:
example.net.            172800  IN      NS      a.iana-servers.net.
example.net.            172800  IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     172800  IN      A       199.43.132.53
b.iana-servers.net.     172800  IN      A       199.43.133.53
```

Ask a .net nameservers.

Go ask them!

```
seed@ubuntu:$ dig @a.iana-servers.net www.example.net

;; QUESTION SECTION:
;www.example.net.               IN      A

;; ANSWER SECTION:
www.example.net.        86400   IN      A       93.184.216.34
```

Ask an example.net nameservers.

Finally got the answer

Web and HTTP

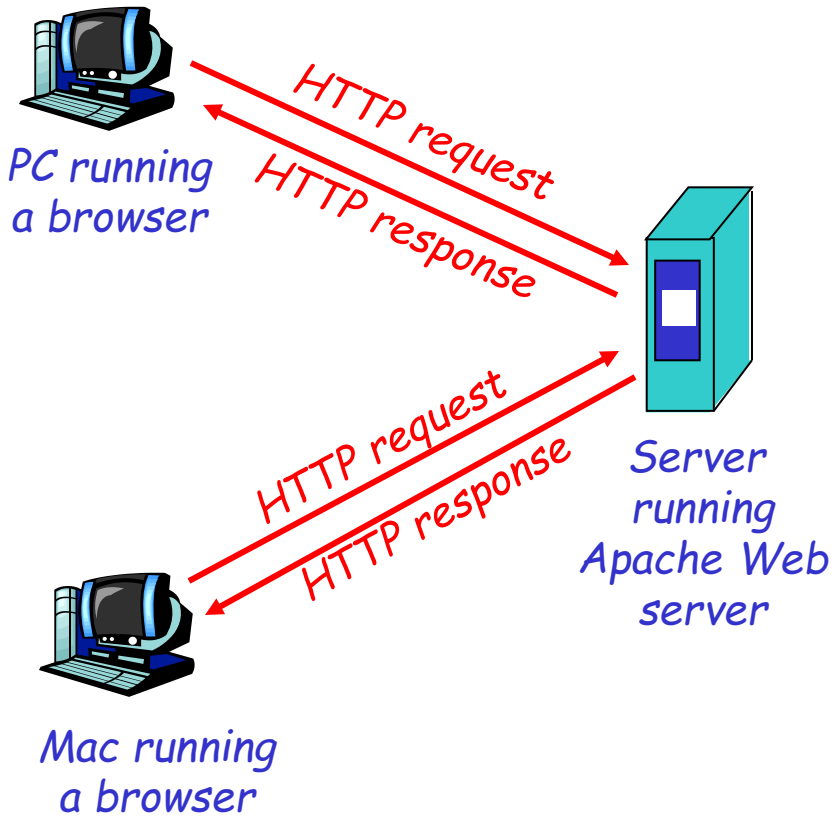
First some jargon

- Web page consists of objects
- Object can be a HTML file, JPEG image, Java applet, audio file,…
- Web page consists of a base HTML-file which usually includes several referenced objects
- Each object is addressable by a URL
- Example URL:

HTTP overview

HTTP: hypertext transfer protocol

- Web's application layer protocol
- client/server model
  - *client:* browser that requests, receives, "displays" Web objects
  - *server:* Web server sends objects in response to requests
- HTTP 1.0: RFC 1945
- HTTP 1.1: continuously updated
- HTTP 3 current version.



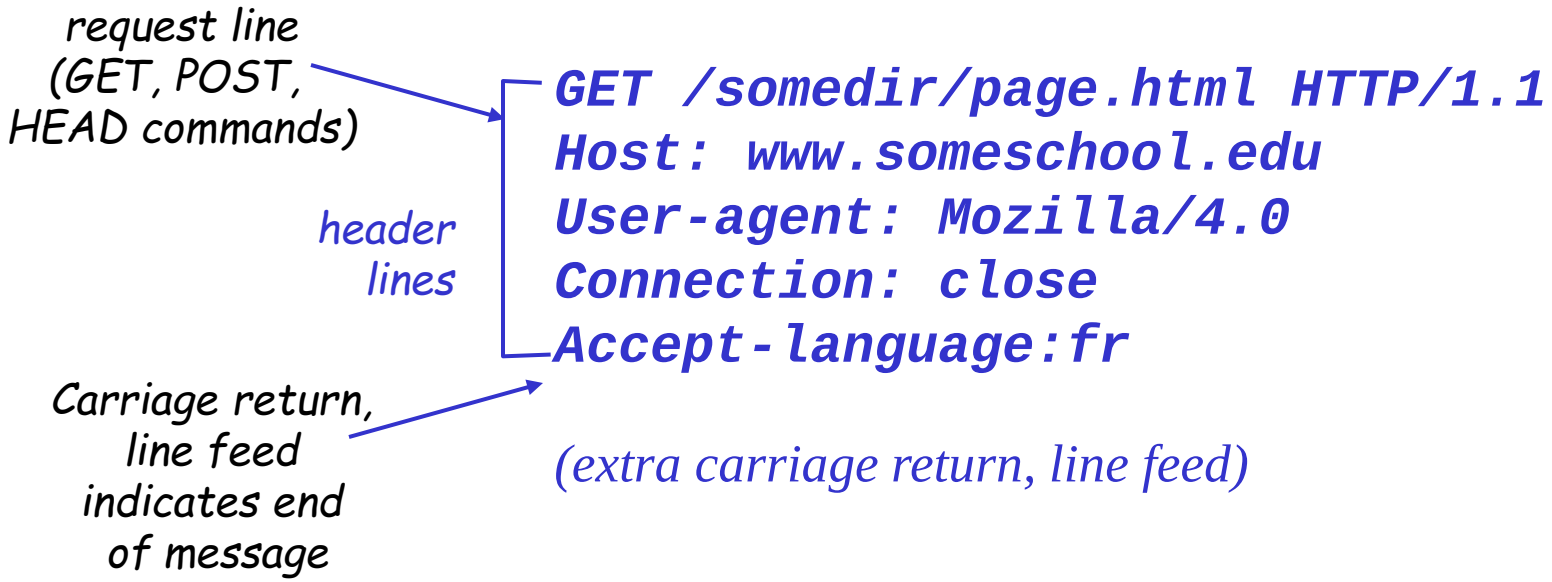
HTTP overview (continued)

Over TCP:

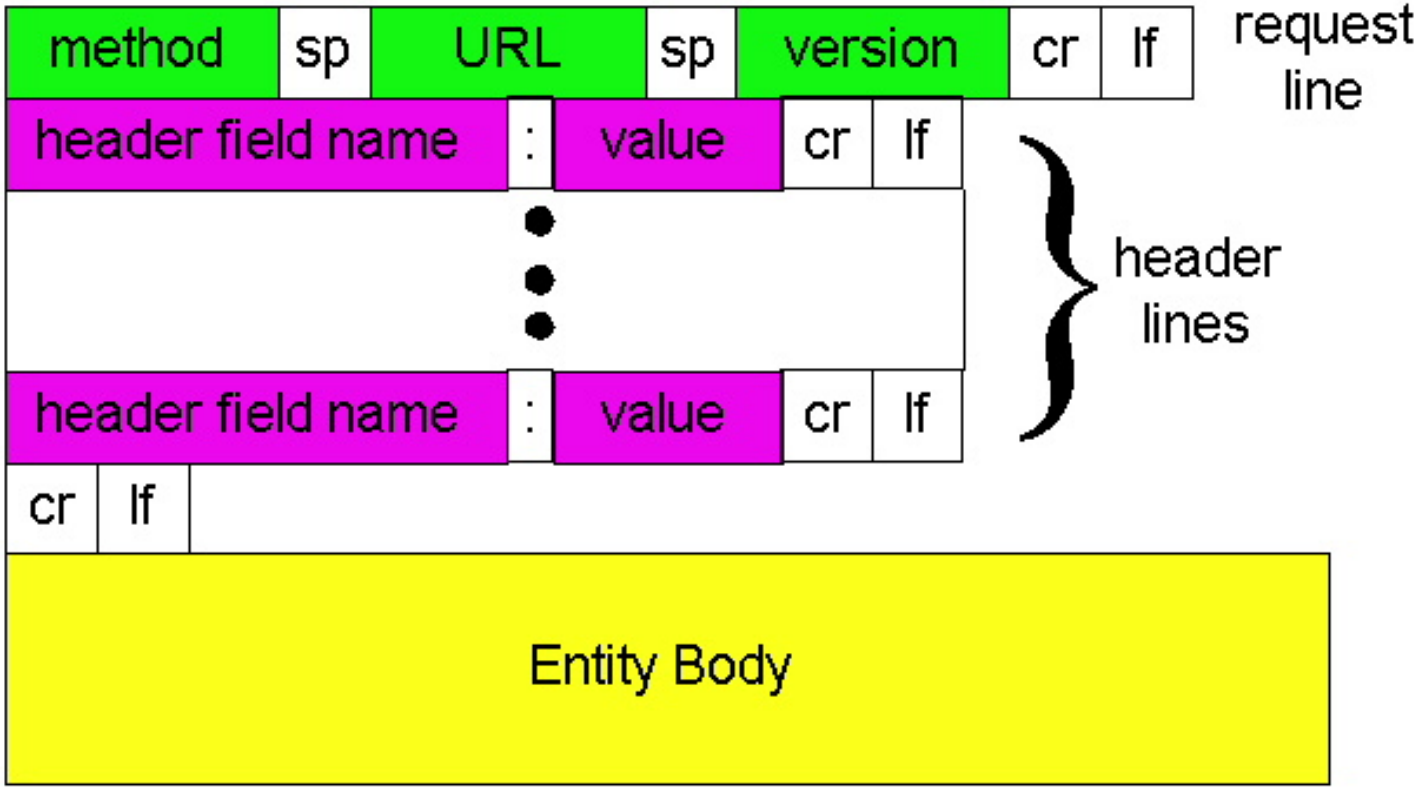
- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

HTTP request message

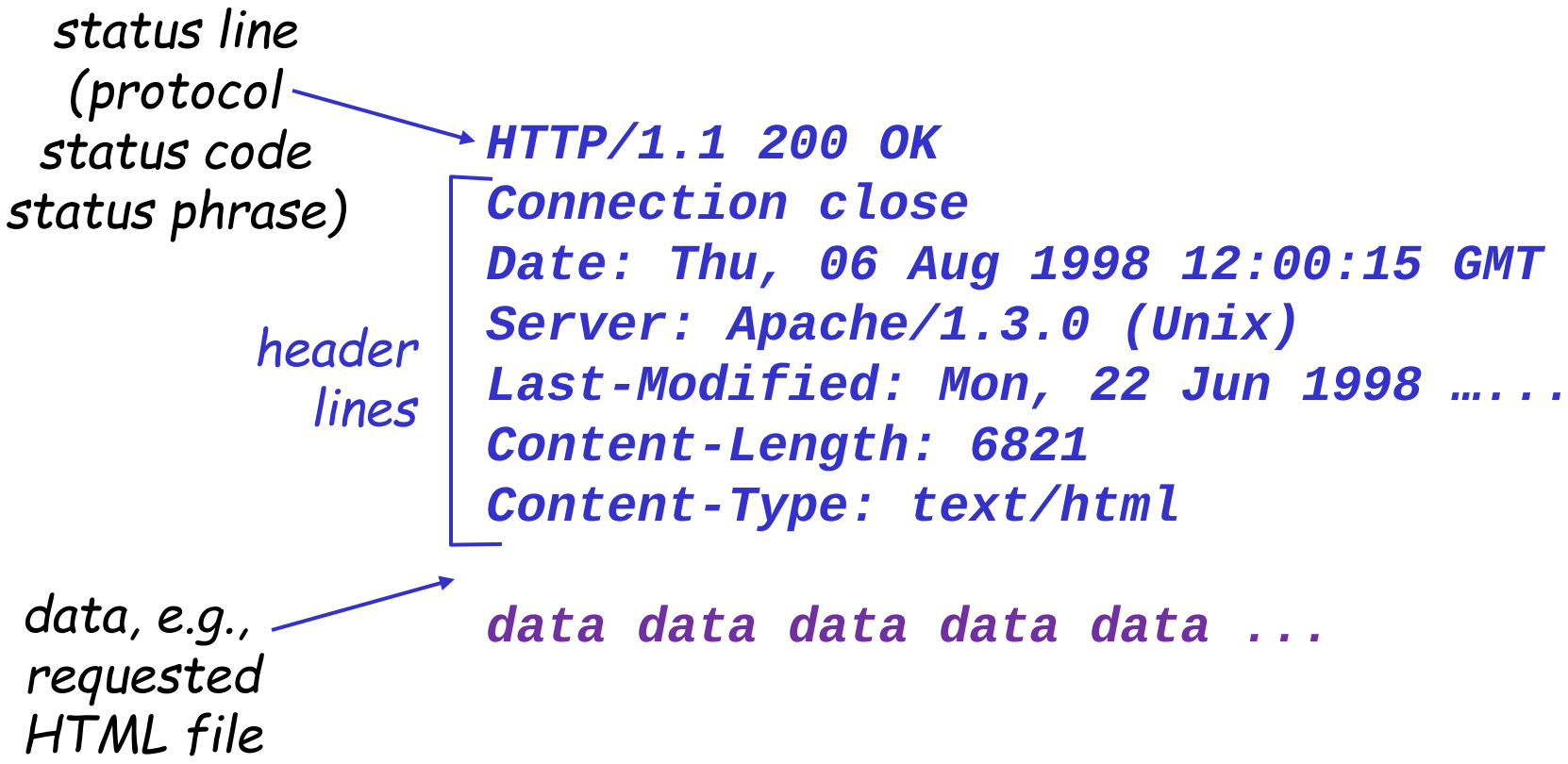
- two types of HTTP messages: *request*, *response*
- HTTP request message:
  - ASCII (human-readable format)



HTTP request message: general format



HTTP response message



Trying out HTTP (client side) for yourself

1. Telnet to your favorite Web server:

```
telnet www.cs.fsu.edu 80
```

Opens TCP connection to port 80 (default HTTP server port) at www.cs.fsu.edu. Anything typed in sent to port 80 at www.cs.fsu.edu

2. Type in a GET HTTP request:

```
GET index.html / HTTP/1.1
Host: www.cs.fsu.edu
```

By typing this in (hit carriage return twice), you send this minimal (but complete) GET request to HTTP server

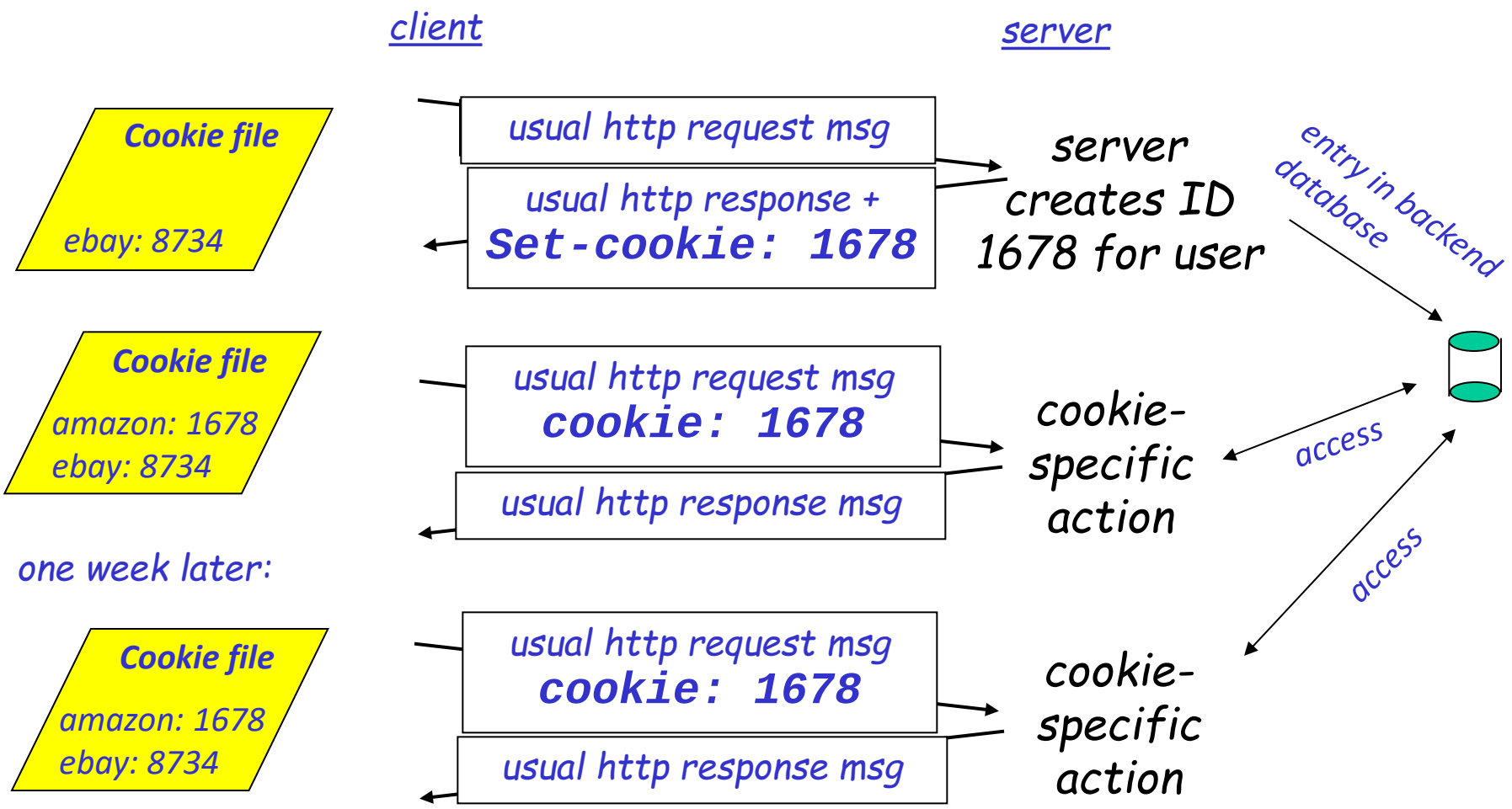
3. Look at response message sent by HTTP server!

```
telnet www.cs.fsu.edu 80
Trying 192.168.23.10...
Connected to www.cs.fsu.edu (192.168.23.10).
Escape character is '^]'.
GET /index.html /HTTP/1.1
Host: www.cs.fsu.edu

HTTP/1.1 200 OK
Date: Wed, 28 Nov 2007 18:34:29 GMT
Server: Apache/2.0.52 (Scientific Linux)
Last-Modified: Mon, 29 Aug 2005 18:02:35 GMT
ETag: "1defce0-29c5-4cd2a4c0"
Accept-Ranges: bytes
Content-Length: 10693
Connection: close
Content-Type: text/html; charset=ISO-8859-1

<html>

<head>
<title>Computer Science @ Florida State University</title>

<base HREF="http://www.cs.fsu.edu/">

<meta   NAME="resource-type"  CONTENT="document">
<meta   NAME="description"    CONTENT="Website for the Computer Science Department
                              at Florida State University">
<meta   NAME="keywords"       CONTENT="Florida State University, Computer Science,
                              Internet2, CS">
<meta   NAME="distribution"   CONTENT="global">
<meta   NAME="author"         CONTENT="Kendal Van Dyke">
```

User-server state: cookies

- HTTP is stateless.
  - two requests are treated independently.
  - Why stateless?
  - What is the problem with a stateless http?
    - E-commence: People buy things by making many requests. Need the ability to bind the requests from the same customer together.
  - Solution: cookies (RFC 2109)
    - Small named string of max 4 kb. Typically has 5 fields:
      - *domain*: where the cookie came from, for example amazon.com
      - *path:* valid subtree of server, usually whole tree /
      - *content*: arbitrary info of type name = value, eg. customerid =4321
      - *expires:* date cookie expires
      - *secure:* need to use SSL/TLS or not

Cookies: keeping "state" (cont.)



Cookies (continued)

What cookies can bring:

- authorization
- shopping carts
- recommendations
- user session state (Web e-mail)

aside

Cookies and privacy:

- *cookies permit sites to learn a lot about you*
- *you may supply name and e-mail to sites*
- *search engines use redirection & cookies to learn yet more*
- *advertising companies obtain info across sites*

- Some issues in HTTP:
  - Mainly due to its popularity
  - Cache support.
    - Insufficient in http/1.0, improved in http/1.1, better now.
    - Intermediate nodes, encoding, etc
  - Dynamically generated date
    - Not reliable in http/1.0, better now.
  - Performance
    - Persistent or non-persistent TCP connection
    - Download the whole file or part of a file
  - User preference
  - Security

Content-Delivery Akami

- With 365,000+ servers, 135+ countries, 4200+ PoPs, 1200+ networks.
- Most likely what happens when a request for a webpage, the original server will reply to the text html file while asking Akamai servers to send the large files.
- Good for the clients, and good for the ISPs because it reduces the upstream traffic.

DNS Redirecting

- First time, the client needs to look up www.cdn.com so it will get the DNS reply from the CDN DNS server.
- The CDN DNS server finds which server best serves this client and returns the ip address of this server, considering the network capacity, the network delay, and the network load.
- Akamai also has to consider where to put the servers.