Application layer

- Electronic Mail
  - SMTP, POP3, IMAP

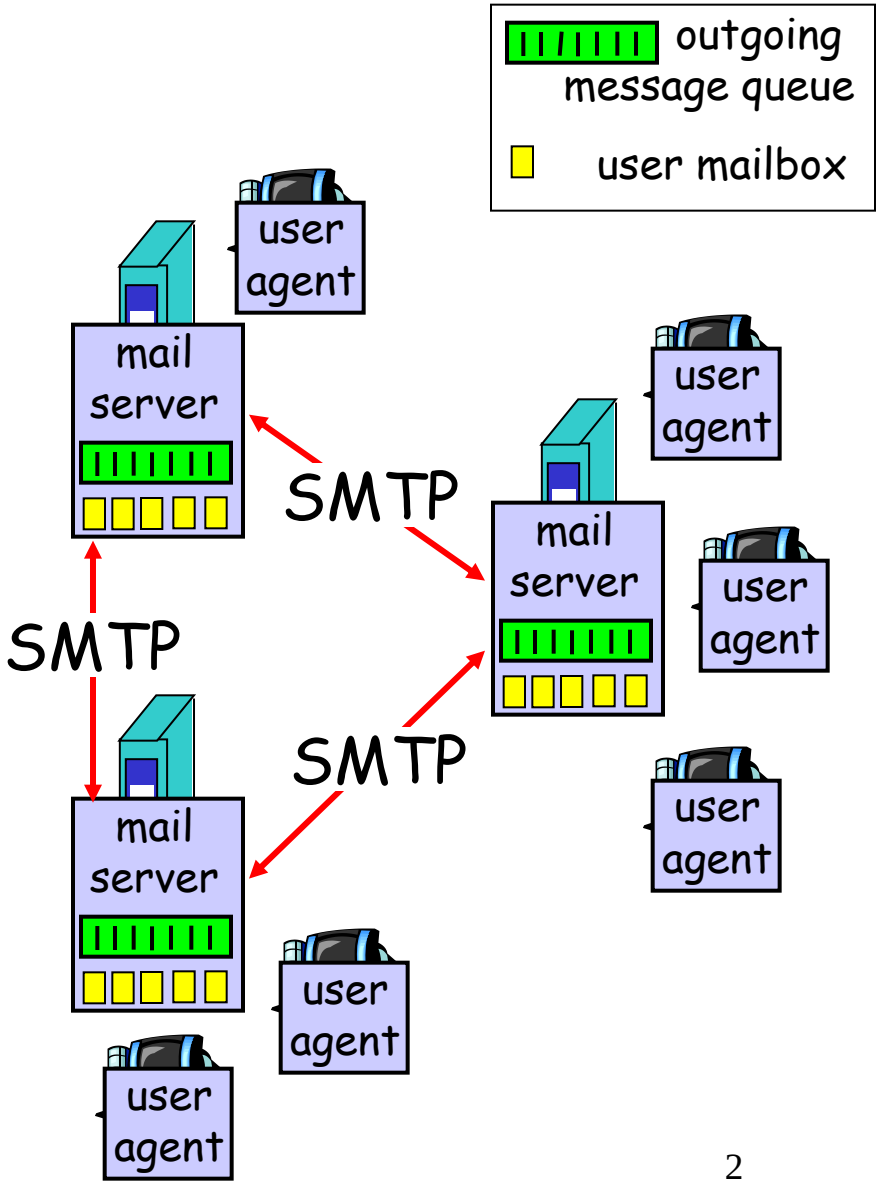
Electronic mail

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

- User Agent

- Is a mail reader
- Able to compose, edit and read mail messages
- e.g., Eudora, Outlook, elm, Netscape Messenger…
- outgoing, incoming messages stored on server



Electronic mail

Mail Servers

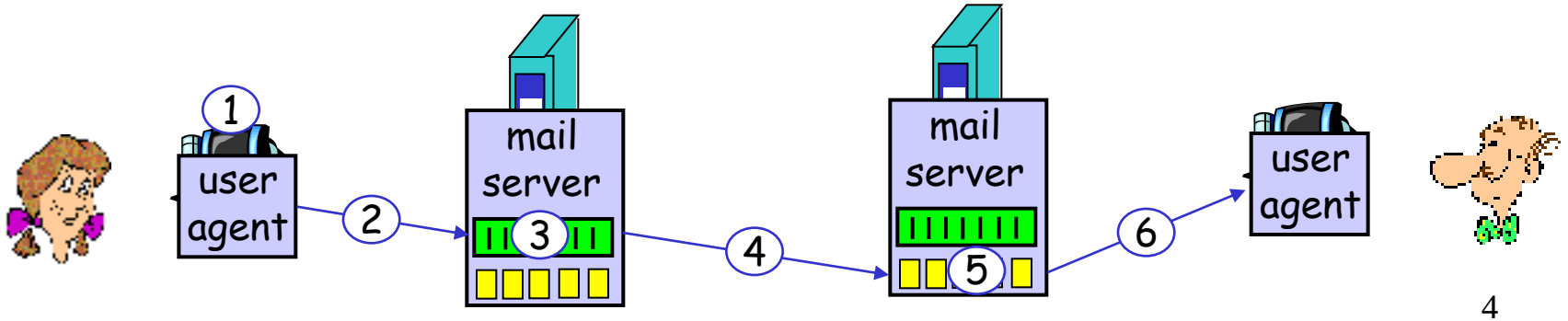
- mailbox contains incoming messages for user
- message queue of outgoing (to be sent) mail messages
- SMTP protocol between mail servers to send email messages
  - client: sending mail server
  - "server": receiving mail server

SMTP [RFC 2821]

- uses TCP to reliably transfer email message from client to server, port 25
- direct transfer: sending server to receiving server
- three phases of transfer
  - handshaking (greeting) transfer of messages closure
- command/response interaction
  - commands: ASCII text
  - response: status code and phrase
- messages must be in 7-bit ASCII

SMTP: scenario

1) Nadine uses U.A. (user agent) to compose message and send "to" samir@yahoo.com
2) Nadine's UA sends message to her mail server; message placed in message queue
3) Client side of SMTP opens TCP connection with Samir's mail server
4) SMTP client sends Nadine's message over the TCP connection
5) Samir's mail server places the message in Samir's mailbox
6) Samir invokes his user agent to read message



LAB: SMTP with telnet

- Connect to email server and send a message.
- Use telnet on port 25
- see 220 reply from server
- enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands
- No need to use a User Agent.

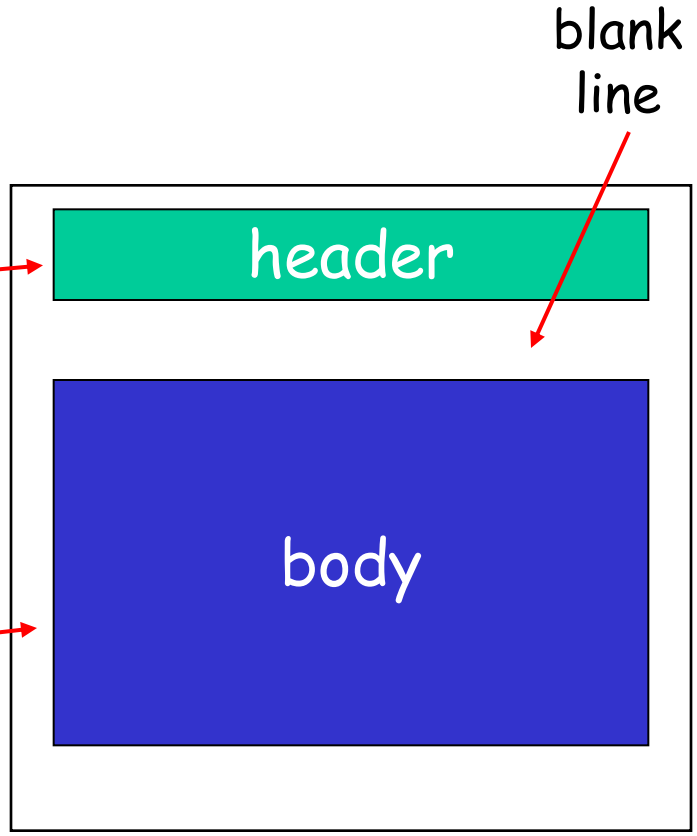
Mail message format

SMTP: protocol for exchanging email msgs

RFC 822: standard for text message format:

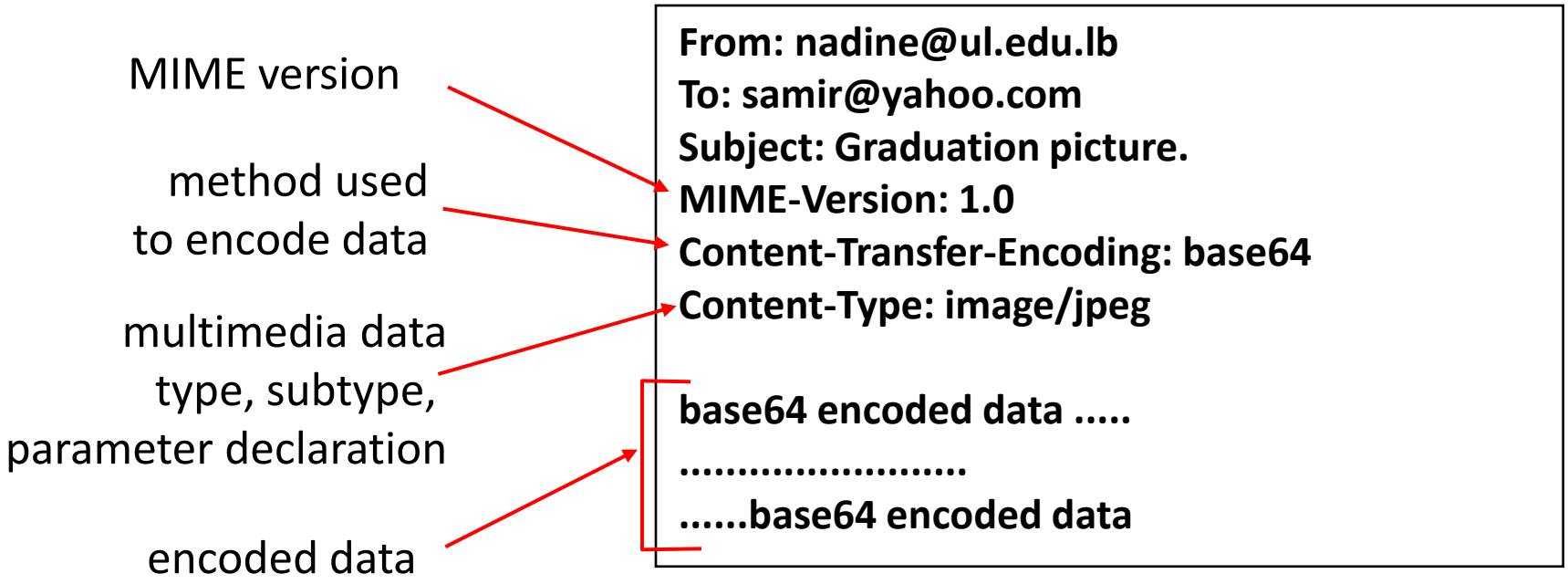
- header lines, e.g.,
  - To:
  - From:
  - Subject:

  different from SMTP commands!
- body
  - the "message", ASCII characters only

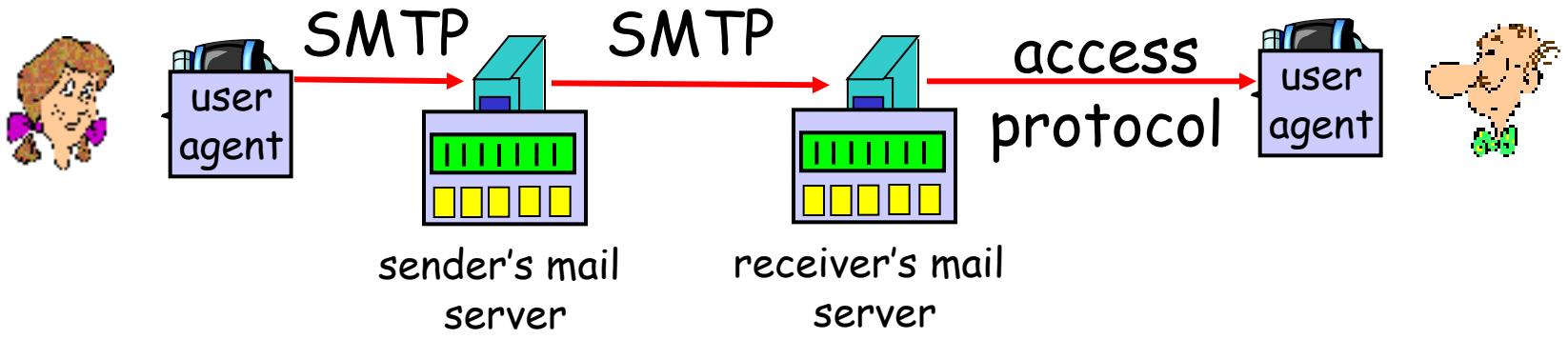


Message format: multimedia extensions

- MIME: multimedia mail extension, RFC 2045, 2056
- additional lines in msg header declare MIME content type



Mail access protocol



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - **POP**: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - **IMAP**: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
  - **HTTP**: Hotmail , Yahoo! Mail, etc.

POP3 protocol

authorization phase

- client commands:
  - **`user`**: declare username
  - **`pass`**: password
- server responses
  - **`+OK`**
  - **`-ERR`**

transaction phase, client:

- **`list`**: list message numbers
- **`retr`**: retrieve message by number
- **`dele`**: delete
- **`quit`**

S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

LAB: POP3 with telnet

- Connect to email server and send a message.
- Use telnet on port 110
- **telnet students.ul.edu.lb 110**
- **User**
- **Pass**
- **List**
- **Try retr, top, dele**

http://smanage.tripod.com/tel.html

POP3

More about POP3

- Previous example uses “download and delete” mode.
  - Nadine cannot re-read e-mail if she changes user client. (because emails are downloaded somewhere else.
- “Download-and-keep” is an offered option in pop3.
- Copies of messages are downloaded.
- Originals remain on the server
- POP3 is stateless across sessions

IMAP

IMAP

- Keep all messages in one place: the server
- Allows user to organize messages in folders
- IMAP keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name

Application layer

- DNS

DNS: Domain name system

People: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address (32 bit) - used for addressing messages
- "name", e.g., ww.yahoo.com - used by humans (easier to memorize than IP)

Q: how to map between IP addresses and name ?

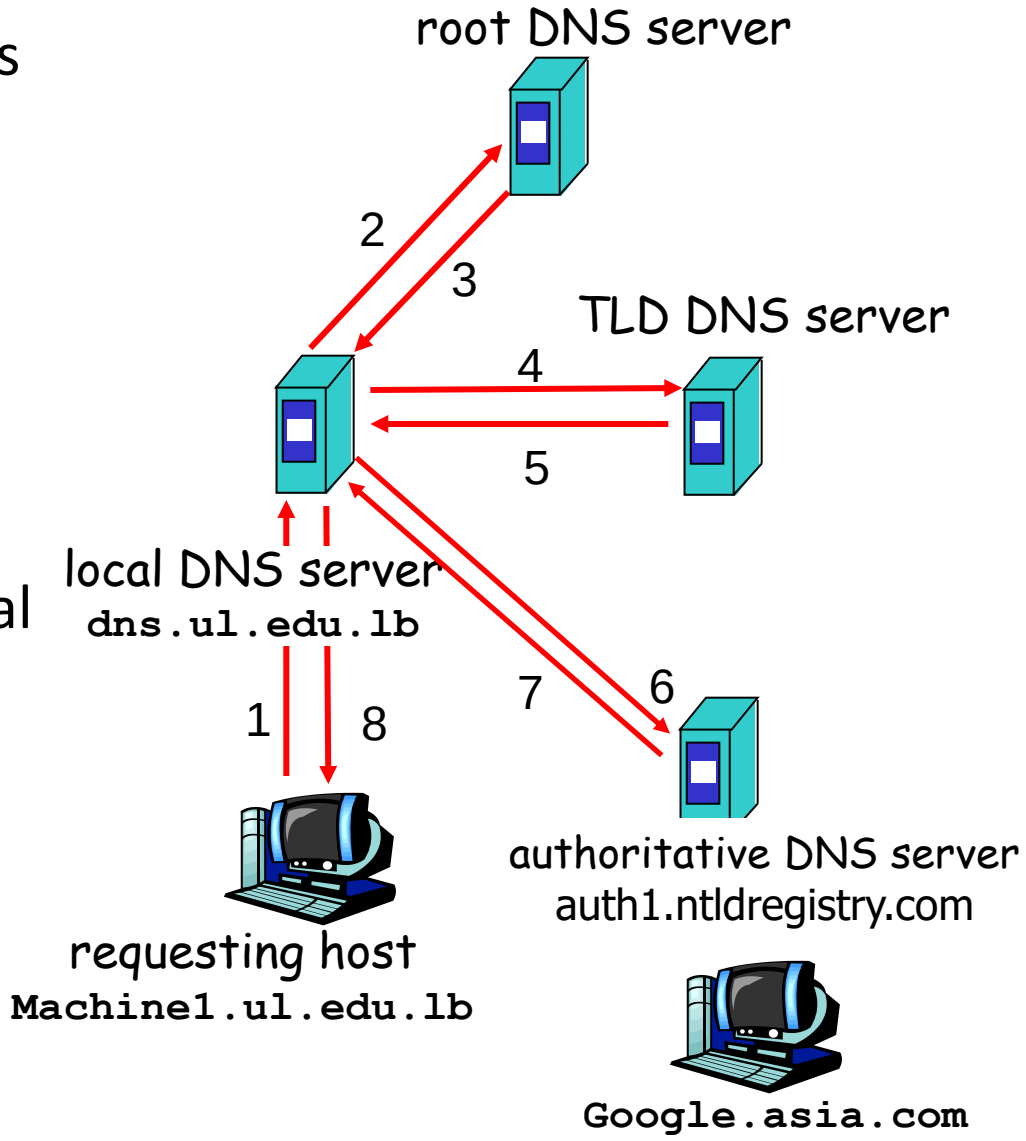
Domain Name System:

- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol* host, routers, name servers to communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol

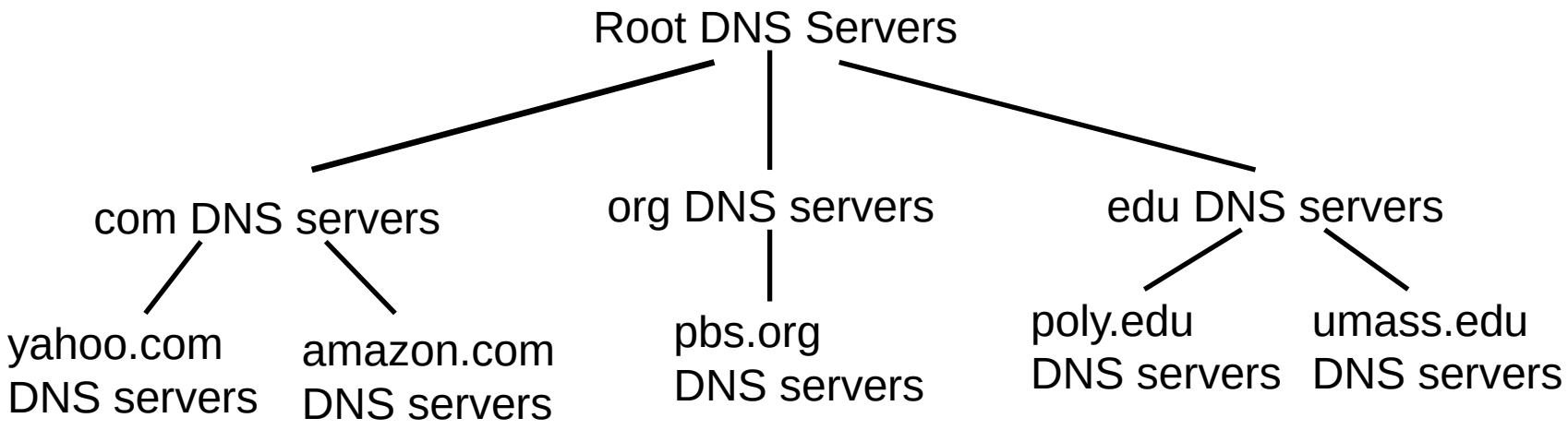
How DNS works

- Host at machine1.ul.edu wants IP address for google.asia.com
- Host sends a "recursion-requested" query request to dns.ul.edu.lb.

- Local DNS server does a "recursive" search.
- This requires contacting several other DNS servers before the final answer is given to host.



Distributed, Hierarchical Database



Client wants IP for www.amazon.com; 1st approx:

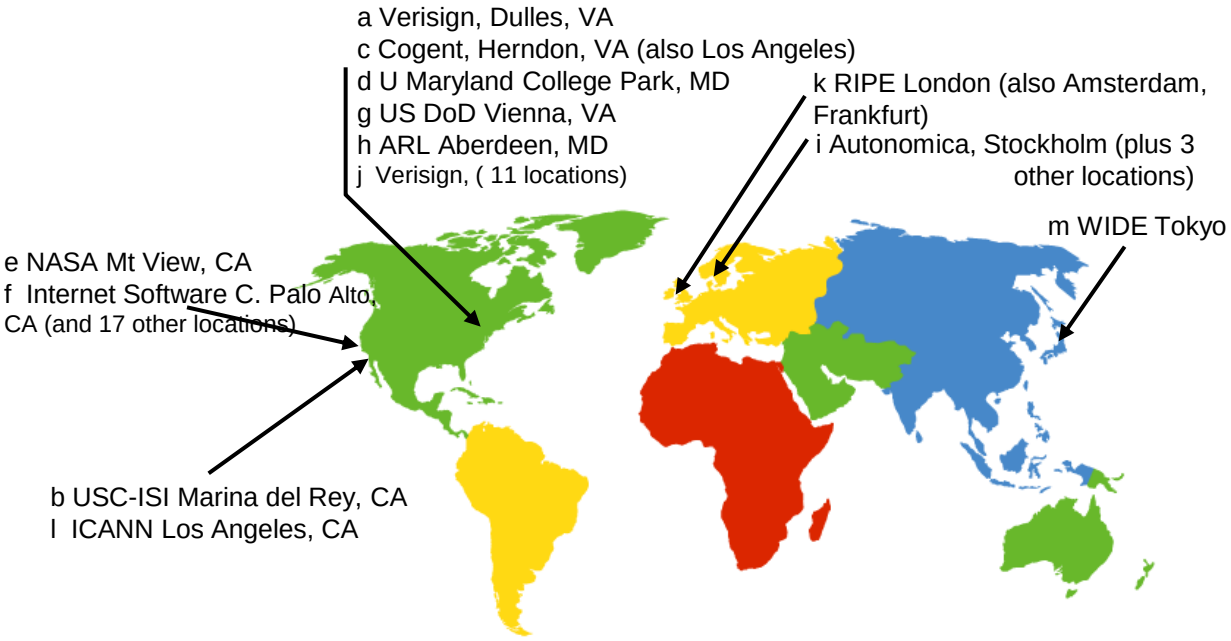
- Client queries a root server to find com DNS server
- Client queries com DNS server to get amazon.com DNS server
- Client queries amazon.com DNS server to get IP address for www.amazon.com

Why not centralize DNS?

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

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



13 root name servers worldwide

TLD and Authoritative Servers

- Top-level domain (TLD) servers: responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp.
  - Network solutions maintains servers for com TLD
  - Educause for edu TLD

- Authoritative DNS servers: organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail).
  - *Can be maintained by organization or service provider*

- Local DNS servers: organization's DNS servers to provide DNS lookups for hosts on the subnet. Maintained by the organization.

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
  - If mapping available, direct response.
  - Otherwise forwards query into hierarchy.
- 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
- update/notify mechanisms under design by IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

LAB: nslookup

nslookup : is a windows tool (nslookup /?)

To get more help type nslookup with no options then type ?

```
C:\>nslookup
Default Server:  UnKnown
Address:  fec0:0:0:ffff::1

> ?
Commands:   (identifiers are shown in uppercase, [] means optional)
NAME            - print info about the host/domain NAME using default server
NAME1 NAME2     - as above, but use NAME2 as server
help or ?       - print info on common commands
set OPTION      - set an option
    all                 - print options, current server and host
    [no]debug           - print debugging information
    [no]d2              - print exhaustive debugging information
    [no]defname         - append domain name to each query
    [no]recurse         - ask for recursive answer to query
    [no]search          - use domain search list
    [no]vc              - always use a virtual circuit
    domain=NAME         - set default domain name to NAME
```

LAB: nslookup

Testing with yahoo.com

1- nslookup yahoo.com

- How many IPs? Why?

Redo the same command every 50 seconds.

- What do you notice? Why?

2- How to discover the hierarchy?

- nslookup norecuse –v yahoo.com

LAB: nslookup

DNS queries from your computer are cashed on your computer

ipconfig /displaydns to display cashed dns.

Test

Connect to yahoo.com, google.com
Open a command prompt window and find out the cashed DNS.

To clear the cashed DNS entries…
ipconfig /flushdns (you need to be administrator)

DNS records

DNS: distributed database storing Resource Records

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
- **value** is canonical name

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

Type=MX

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

DNS protocol, messages

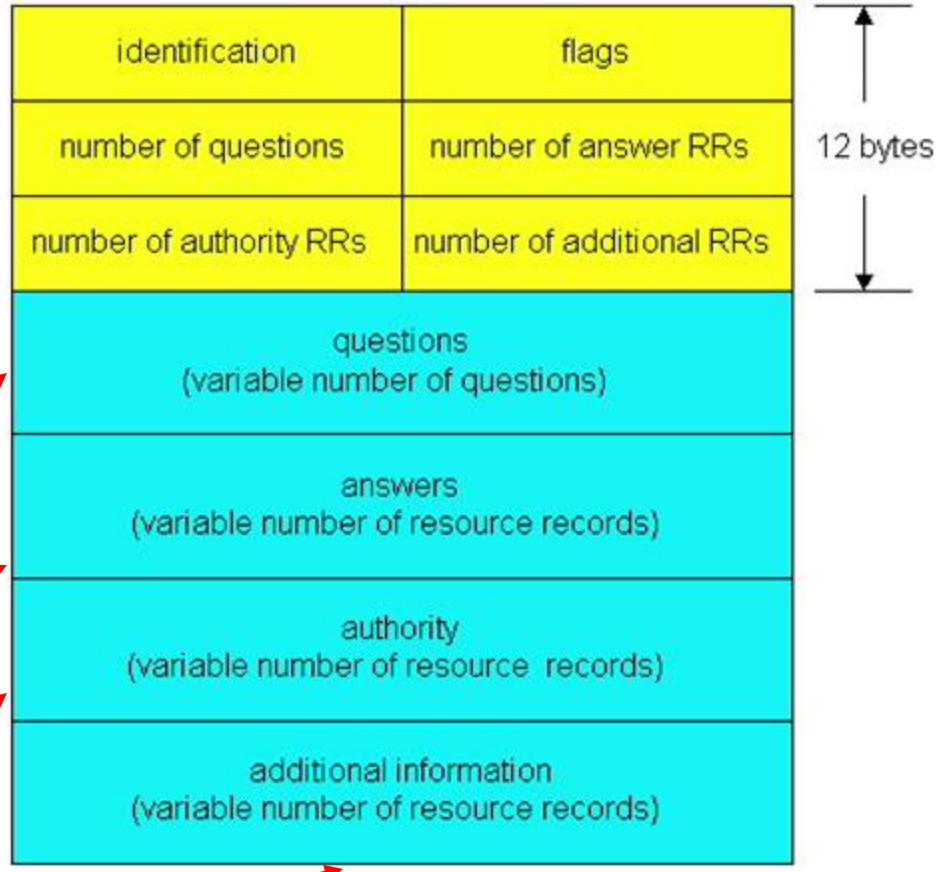
- DNS protocol : *query* and *reply* messages, both with same *message format*

Message header

identification: 16 bit # for query, reply to query uses same #

flags:

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



Name, type fields for a query

RRs in response to query

records for authoritative servers

additional "helpful" info that may be used

Port numbers

Destination port number	Abbreviation	Definition
20	FTP Data	File transfer protocol (for data)
21	FTP Control	File transfer proto
23	Telnet	Teletype Network
25	SMTP	Simple mail transfer protocol
53	DNS	Domain name service
67	DHCP v4 client	Domain host configuration protocol (client)
68	DHCP v4 server	Domain host configuration protocol (server)
69	TFTP	Trivial File transfer protocol
80	HTTP	Hypertext transfer protocol
110	POP3	Post office protocol (version 3)
137	NBNS	Microsoft NetBIOS Name Service
143	IMAP4	Internet Message Access Protocol (version 4)
161	SNMP	Simple Network Management Protocol
443	HTTPS	Hypertext transfer protocol secure