BRAC UNIVERSITY

Inspiring Excellence

Application Layer (Electronic Mail &DNS)

Lecture 4 | Part 1 of 4 | CSE421 – Computer Networks

Department of Computer Science and Engineering
School of Data & Science

Objectives

- Principles of network applications
- Web and HTTP
- **Electronic mail**
  - **SMTP, POP3, IMAP**
- DNS
- P2P applications
- Video streaming and content distribution networks
- Socket programming with UDP and TCP

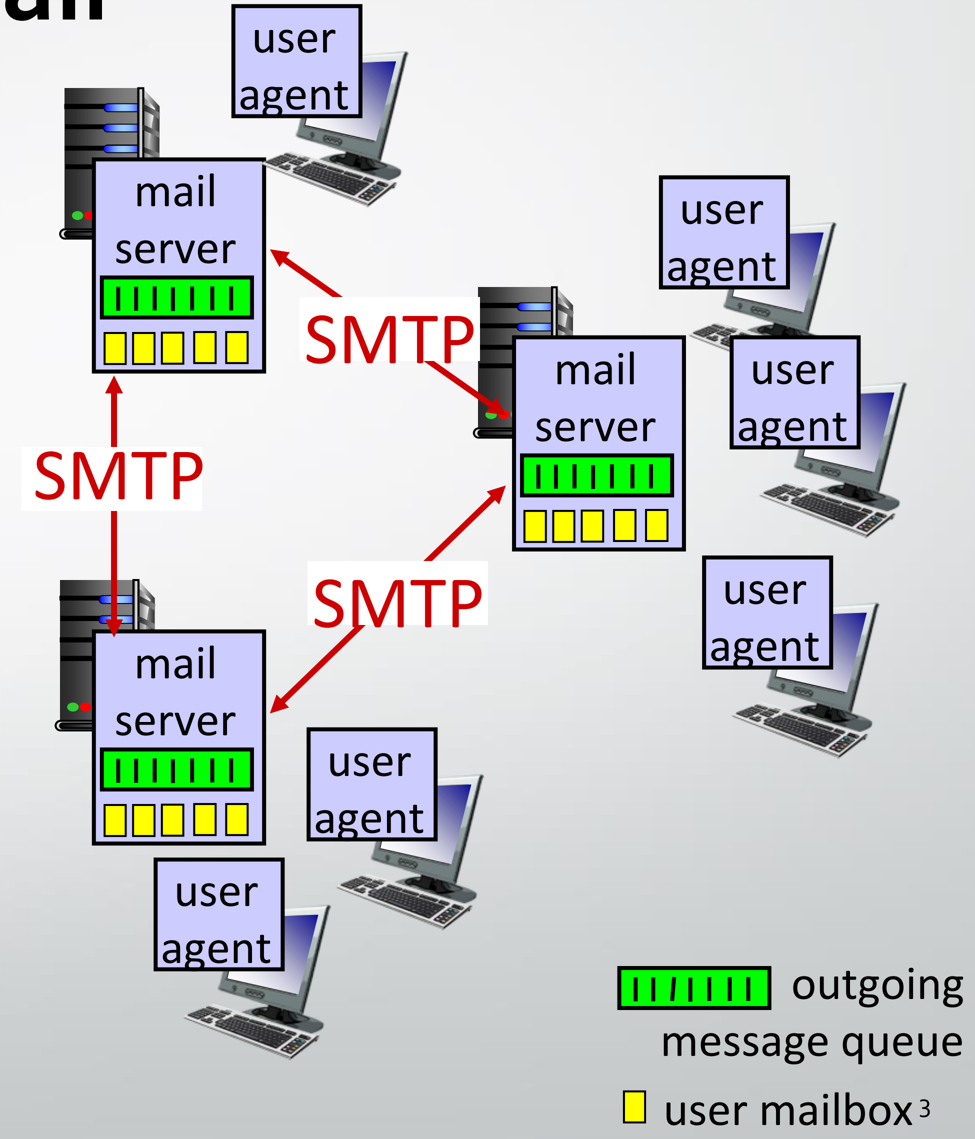
Electronic mail

Three major components:

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

User Agent

- a.k.a. “mail reader”
- composing, editing, reading mail messages
- e.g., Outlook, iPhone mail client
- outgoing, incoming messages stored on server



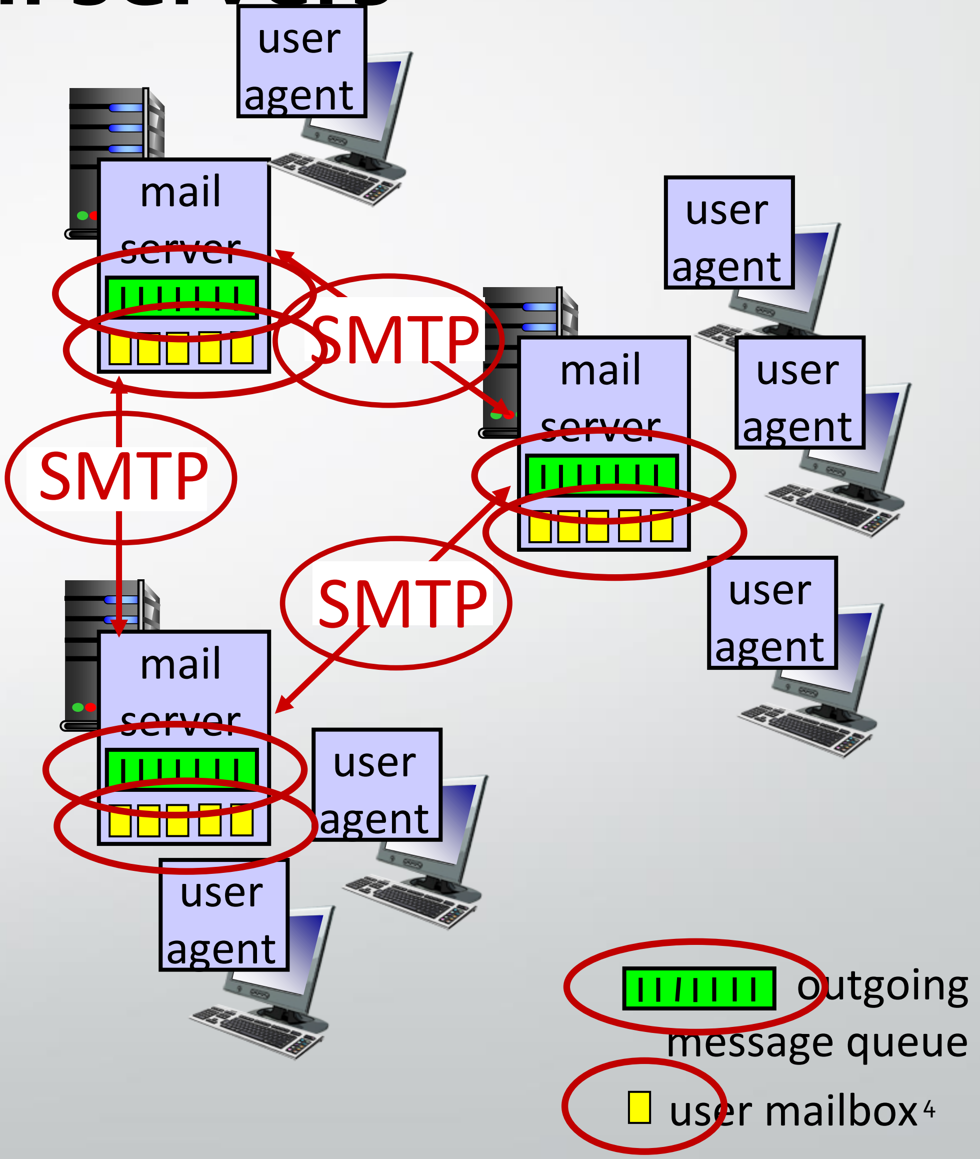
Electronic mail: mail servers

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

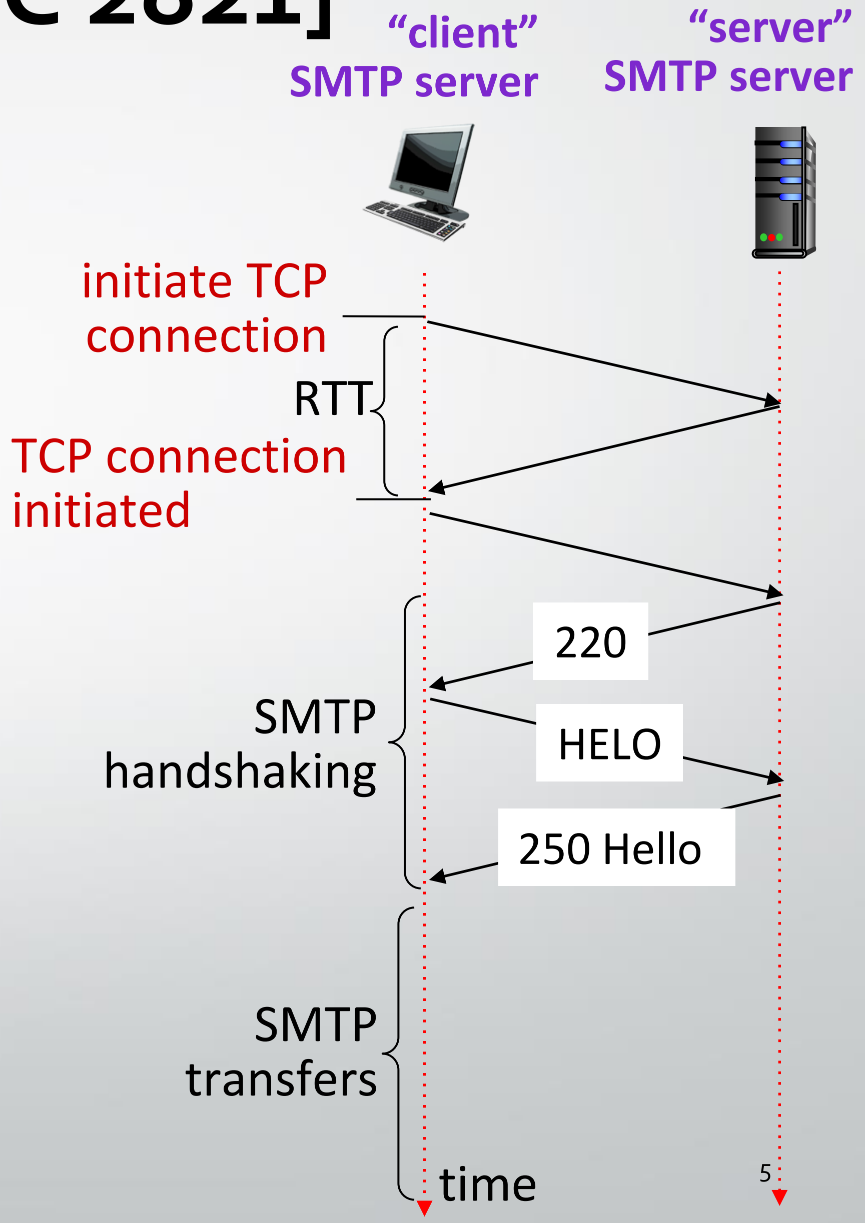


Electronic Mail: SMTP [RFC 2821]

- uses **TCP** to reliably transfer email message from client to server, port 25
  - direct transfer: sending server (acting like client) to receiving server

- **three phases of transfer**
  - SMTP handshaking (greeting)
  - SMTP transfer of messages
  - SMTP closure

- command/response interaction (like HTTP)
  - commands: ASCII text
  - response: status code and phrase



Scenario: Alice sends e-mail to Bob

1) Alice uses UA to compose e-mail message “to” bob@someschool.edu

2) Alice’s UA sends message to her mail server using SMTP; message placed in message queue

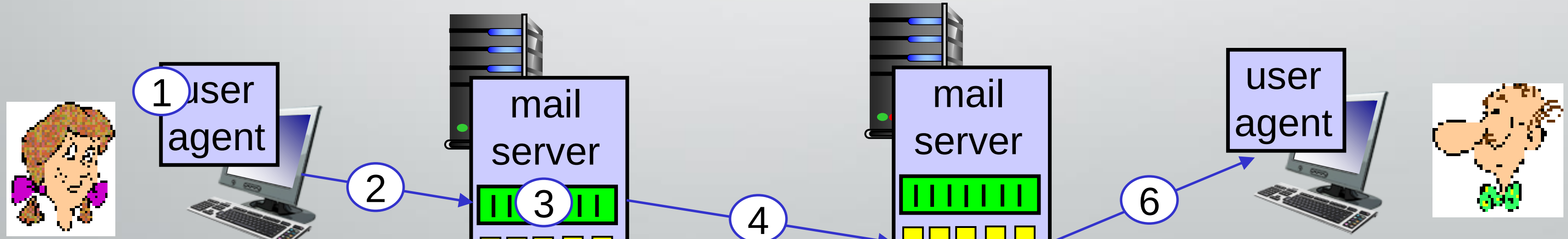
3) client side of SMTP at mail server opens TCP connection with Bob’s mail server

4) SMTP client sends Alice’s message over the TCP connection

5) Bob’s mail server places the message in Bob’s mailbox

6) Bob invokes his user agent to read message

**if connection fails, it keeps retrying for few days



Alice’s mail server

Bob’s mail server

Sample SMTP interaction

```
S: 220 hamburger.edu
```

SMTP: final words

- SMTP uses persistent connections
- SMTP requires message (header & body) to be in 7-bit ASCII
- SMTP server uses CRLF.CRLF (\r\n.\r\n) to determine end of message

Comparison with HTTP:

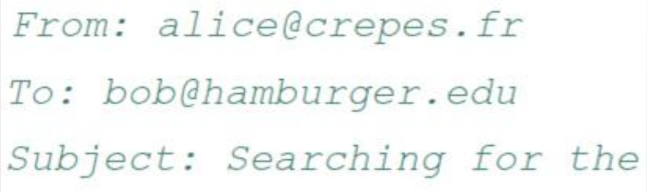
- HTTP: pull; SMTP: push
- HTTP: Server to client; vice versa
- SMTP: server to server
- both have ASCII command/response interaction, status codes
- HTTP: each object encapsulated in its own response message
- SMTP: multiple objects sent in multipart message

Mail message format

SMTP: protocol for exchanging email messages

RFC 822: standard for text message format:

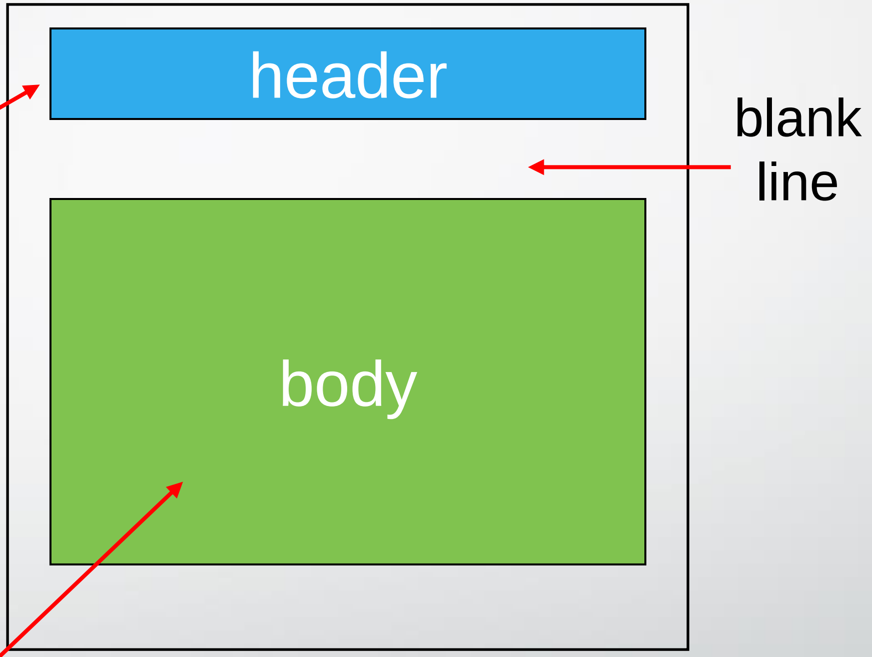
- header lines, e.g.,



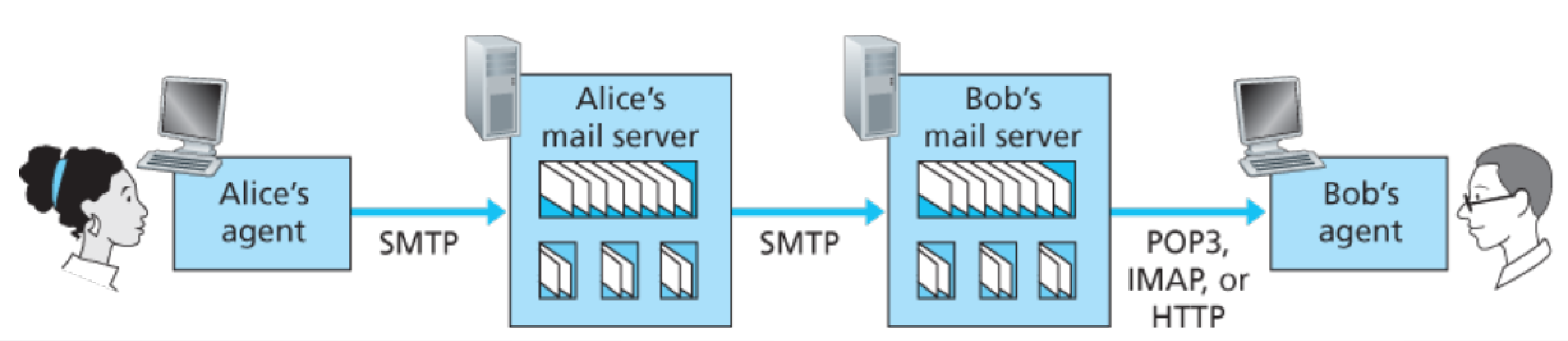
```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Searching for the
```

different from SMTP MAIL FROM, RCPT TO: commands!

- Body: the "message"
  - ASCII characters only



Mail access protocols



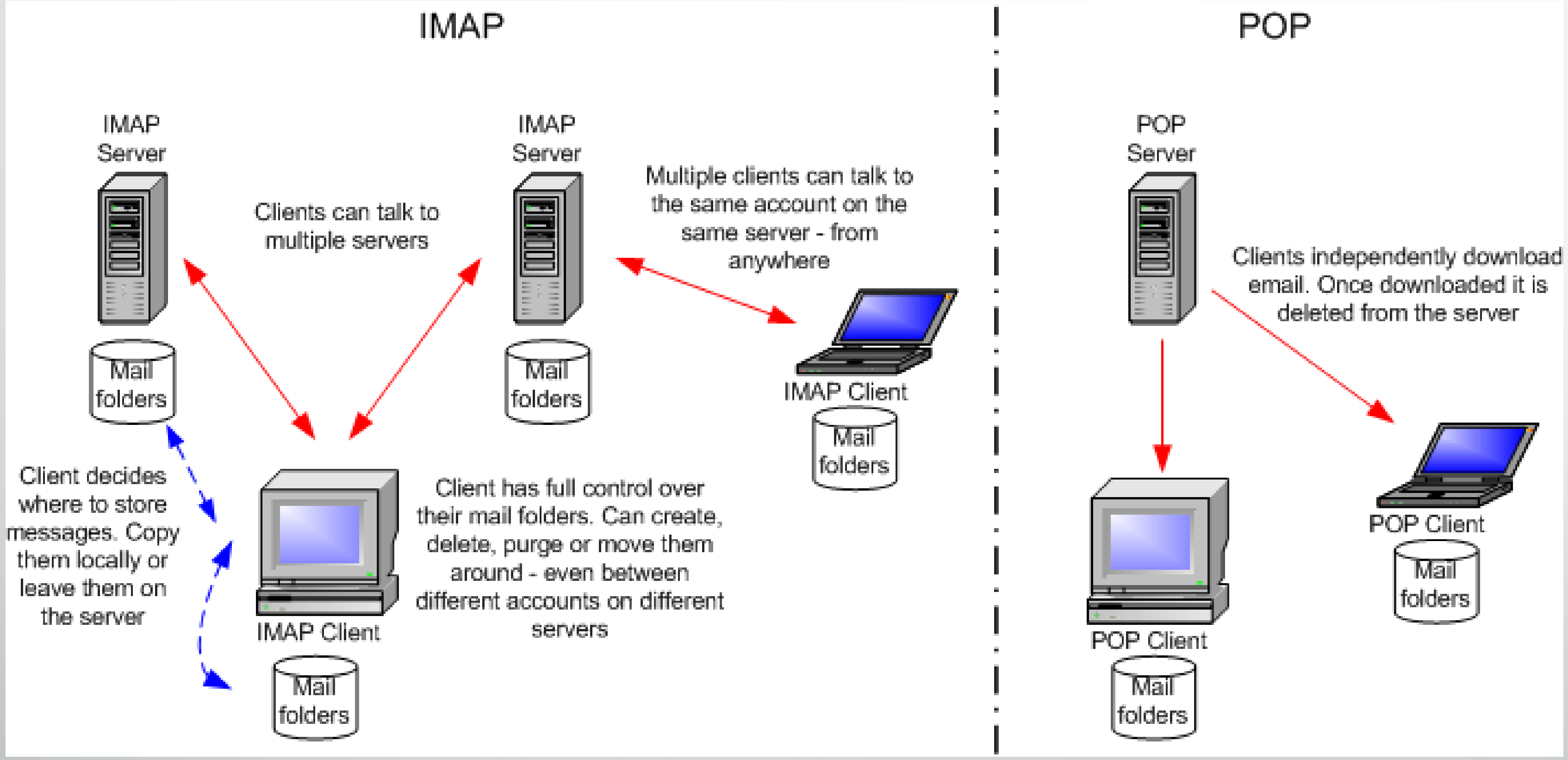
- SMTP: delivery/storage to receiver's server
- mail access protocol: retrieval from server
    - POP: Post Office Protocol [RFC 1939]: authorization, download
    - IMAP: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored messages on server
    - HTTP: Web based(Gmail, Hotmail, Yahoo! Mail, etc.)

POP3 and IMAP

Features	POP3	IMAP
Name	Post Office Protocol	Internet Message Access Protocol
Mail Location	Mail downloaded at the local workstation and deleted from the server. **	Keeps all mails in one place: at the server
Accessing Mail	Mail can only be accessed using a single device at a time when using POP3.	Messages can be accessed via IMAP on a variety of devices
Update	POP3 does not allow users to create, delete, or modify mailboxes on the mail server.	IMAP allows the user to create, delete, or update mailboxes on the mail server, as well as create a folder hierarchy of mailboxes.
Readability	Once the message has been downloaded, we can only read it.	Before we finish the download, we can read the message in part.
Virus	Mail kept in workstation, vulnerable to any virus	Mails kept in server, less susceptible to virus
Port Number	110	143

**POP3 “download-and-keep”: copies of messages on different clients

POP3 and IMAP



IMAP
POP
IMAP Server
IMAP Server
Clients can talk to multiple servers
Multiple clients can talk to the same account on the same server - from anywhere
Mail folders
Mail folders
IMAP Client
Mail folders
Client decides where to store messages. Copy them locally or leave them on the server
Client has full control over their mail folders. Can create, delete, purge or move them around - even between different accounts on different servers
IMAP Client
Mail folders
POP Server
Clients independently download email. Once downloaded it is deleted from the server
POP Client
Mail folders
POP Client
Mail folders

Source : https://www.howtogeek.com/99423/email-whats-the-difference-in-pop3-imap-and-exchange/

Chapter 2: Outline

- Principles of network applications
- Web and HTTP
- Electronic mail
    - SMTP, POP3, IMAP
- **DNS**
- P2P applications
- Video streaming and content distribution networks
- Socket programming with UDP and TCP

DNS: domain name system

People: many identifiers:

- NID, 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"

DNS: How does it work?

1. The user machine runs the client side of the DNS application.
2. The browser extracts the hostname, www.someschool.edu, from the URL and passes the hostname to the client side of the DNS application.
3. The DNS client sends a query containing the hostname to a DNS server.
4. The DNS client eventually receives a reply, which includes the IP address for the hostname.
5. Once the browser receives the IP address from DNS, it can initiate a TCP connection to the
   HTTP server process located at port 80 at that IP address

DNS: services, structure

DNS services

- Hostname to IP address translation *DNS Cache
- Host aliasing
  - Canonical – complicated name!
  - Alias names – give the canonicals a shorter name!
    - Can be used to get the canonical name
- Mail server aliasing
  - Web and Mail hostname can be same
- Load distribution:
  - Replicated Web/Mail servers: many IPs correspond to one name
  - Follows round robin method when distributing IP to user

Why not centralize DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Maintenance

AND doesn‘t scale!

Thinking about the DNS

humongous distributed database:

- ~ billion records, each simple

handles many *trillions* of queries/day:

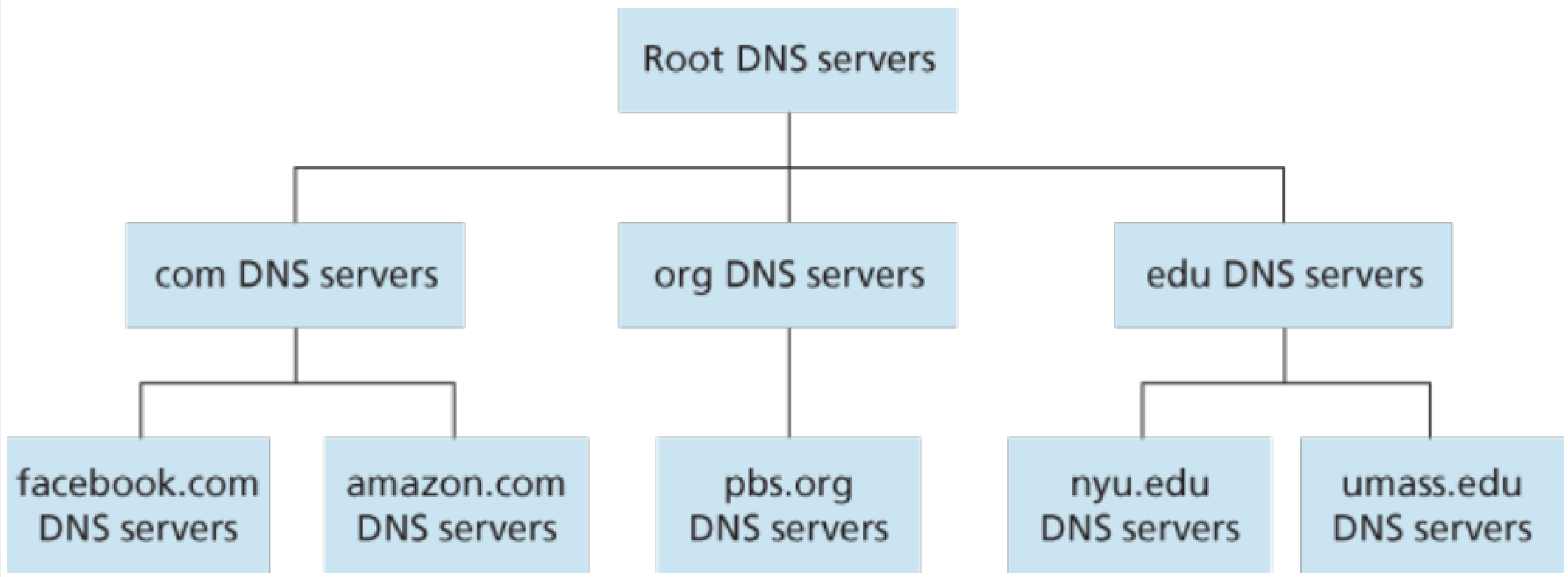
- *many* more reads than writes
- *performance matters:* almost every Internet transaction interacts with DNS - msecs count!

organizationally, physically decentralized:

- millions of different organizations responsible for their records

"bulletproof": reliability, security

DNS: a distributed, hierarchical database

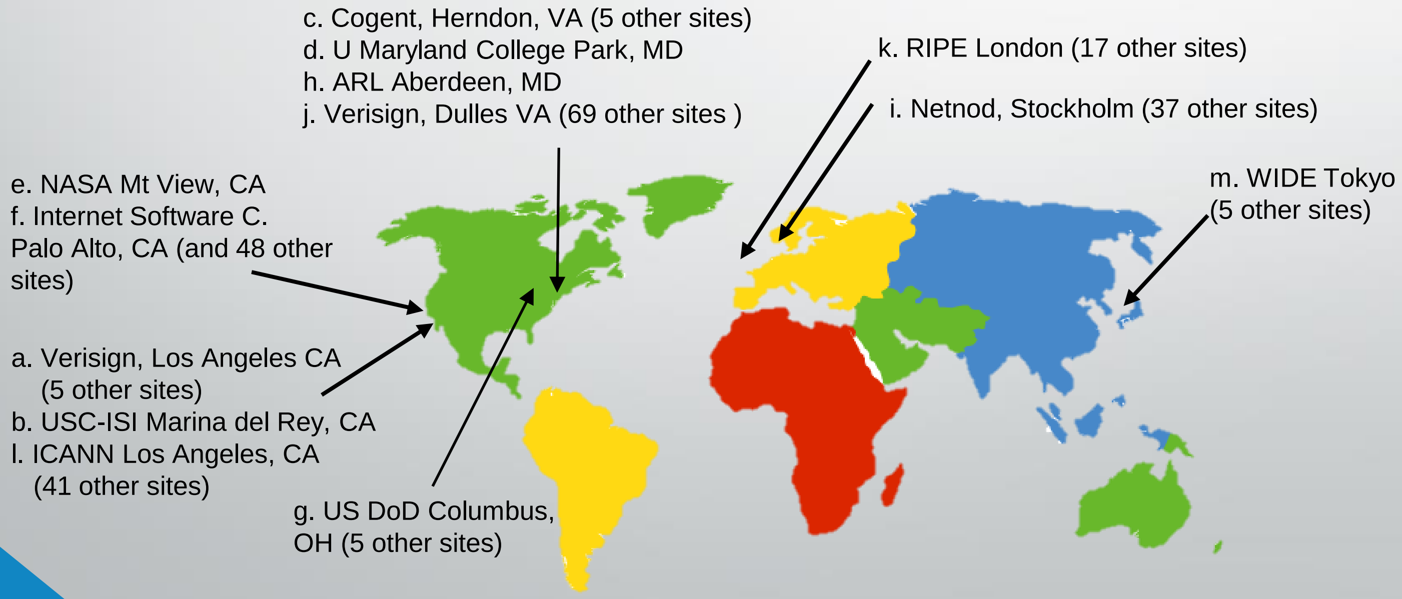


Client wants IP for www.amazon.com; 1st approximation:

- Client queries root server to find the IP of .com DNS server
- Client queries .com DNS server to get IP of an authoritative server of amazon.com.
- Client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: root name servers

- official, contact-of-last-resort by name servers that can not resolve name
- *incredibly important* Internet function
  - Internet couldn't function without it!
  - DNSSEC – provides security (authentication, message integrity)
- ICANN (Internet Corporation for Assigned Names and Numbers) manages root DNS domain

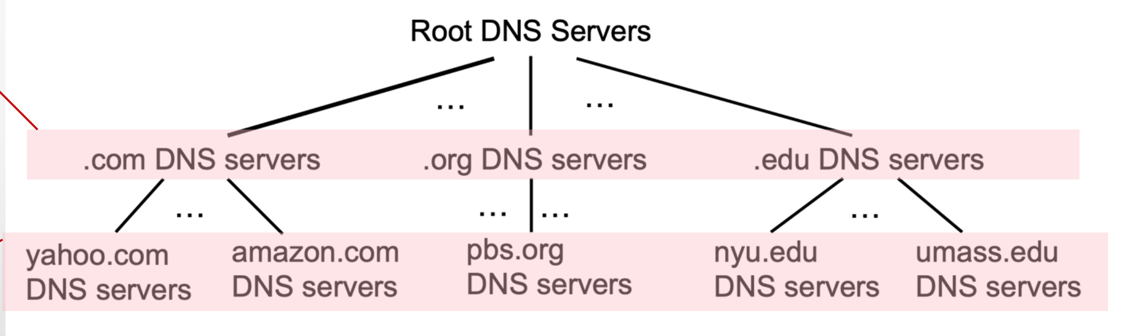

c. Cogent, Herndon, VA (5 other sites)
d. U Maryland College Park, MD
h. ARL Aberdeen, MD
j. Verisign, Dulles VA (69 other sites)
k. RIPE London (17 other sites)
i. Netnod, Stockholm (37 other sites)
m. WIDE Tokyo (5 other sites)
e. NASA Mt View, CA
f. Internet Software C. Palo Alto, CA (and 48 other 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 (~200 servers in US)

Top-Level Domain, and authoritative servers

Top-Level Domain (TLD) servers:

- responsible for .com, .org, .net, .edu, .aero, .jobs, .museums, and all top-level country domains, e.g.: .cn, .uk, .fr, .ca, .jp
- Network Solutions: authoritative registry for .com, .net TLD
- Educause: .edu TLD



authoritative DNS servers:

- organization’s own DNS server(s), providing authoritative hostname to IP mappings for organization’s named hosts
- can be maintained by organization or service provider

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

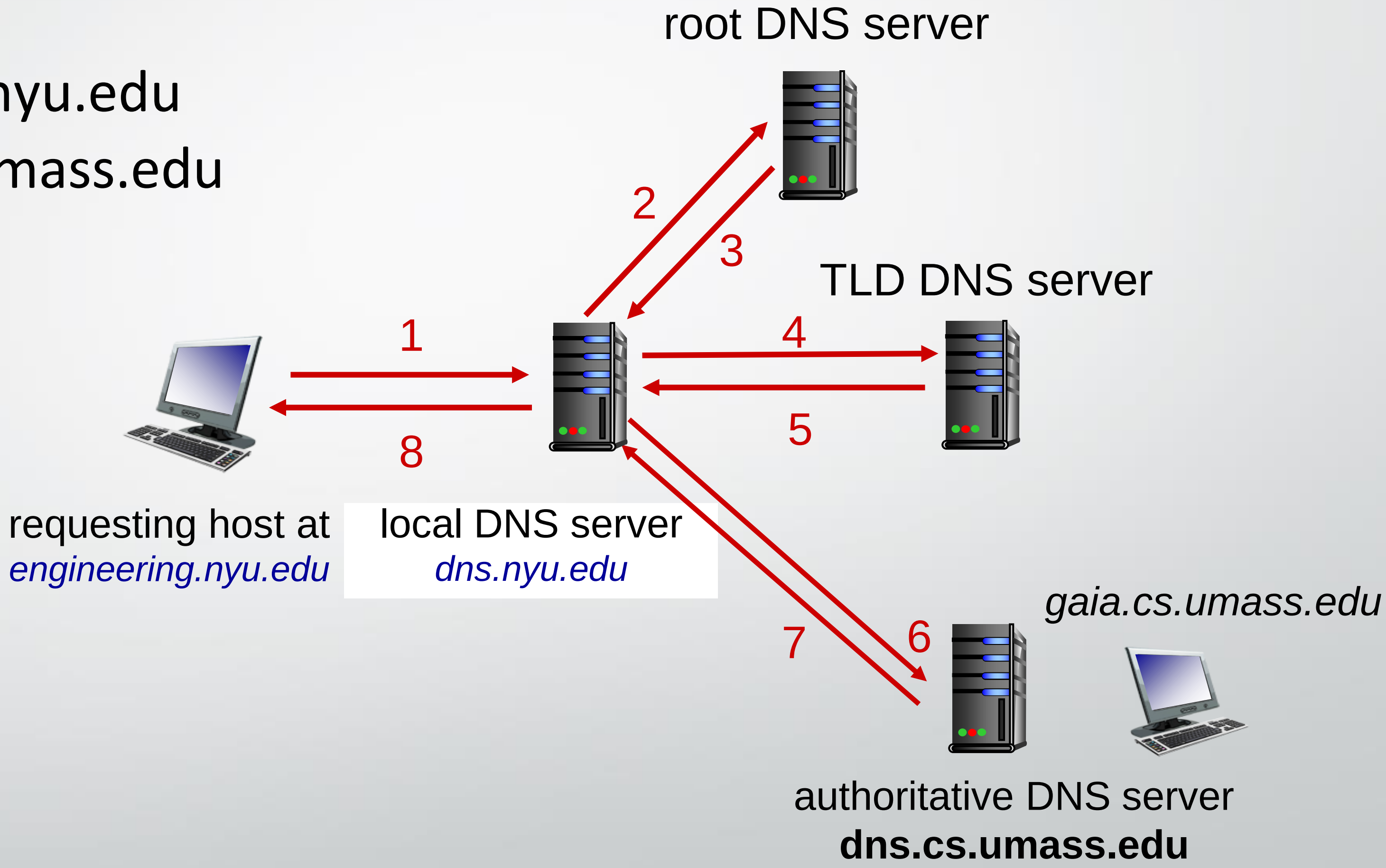
- each ISP has local DNS name server; to find yours:
  - MacOS: `% scutil --dns`
  - Windows: `>ipconfig /all`

DNS name resolution: iterated query

Example: host at engineering.nyu.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"

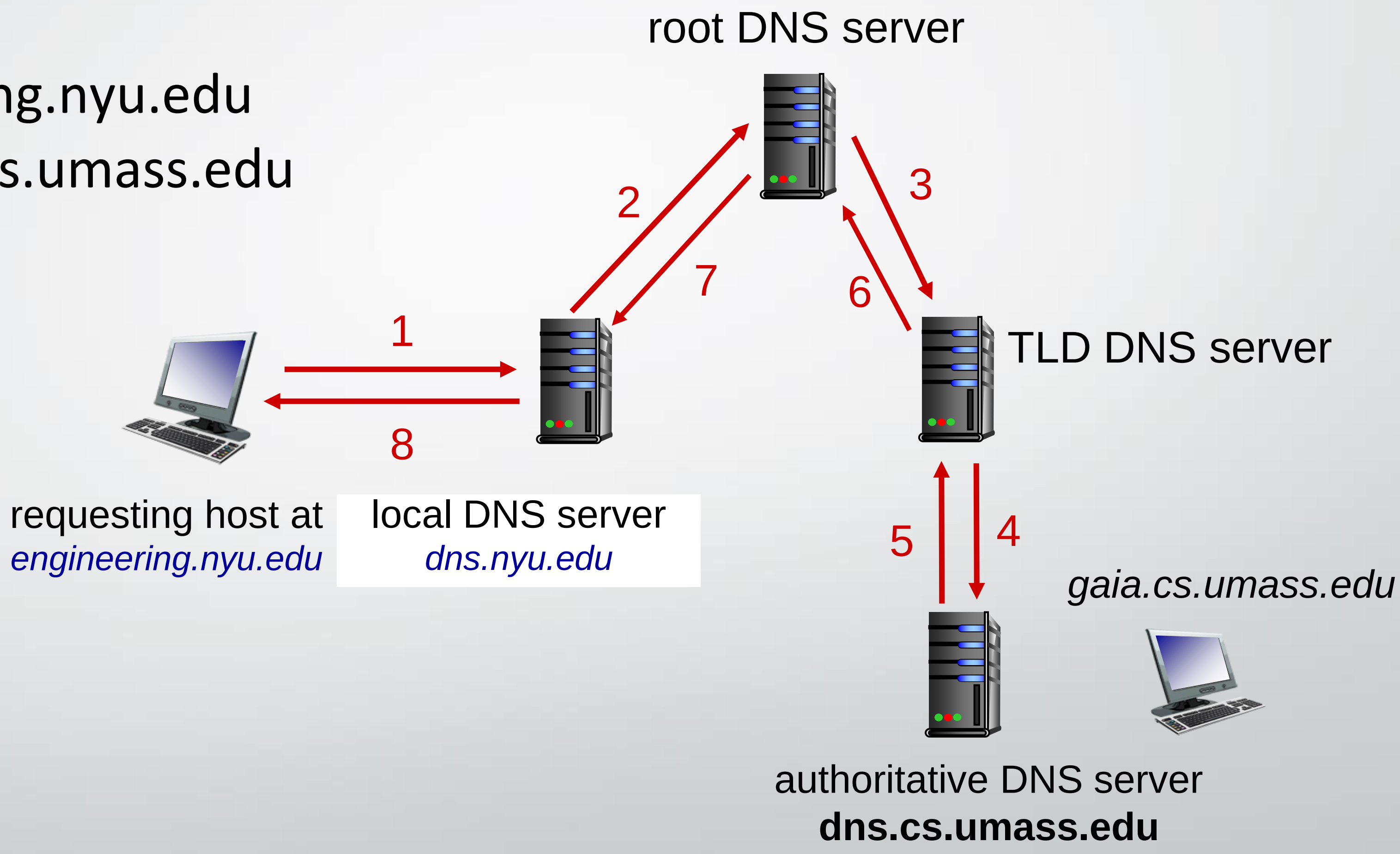


DNS name resolution: recursive query

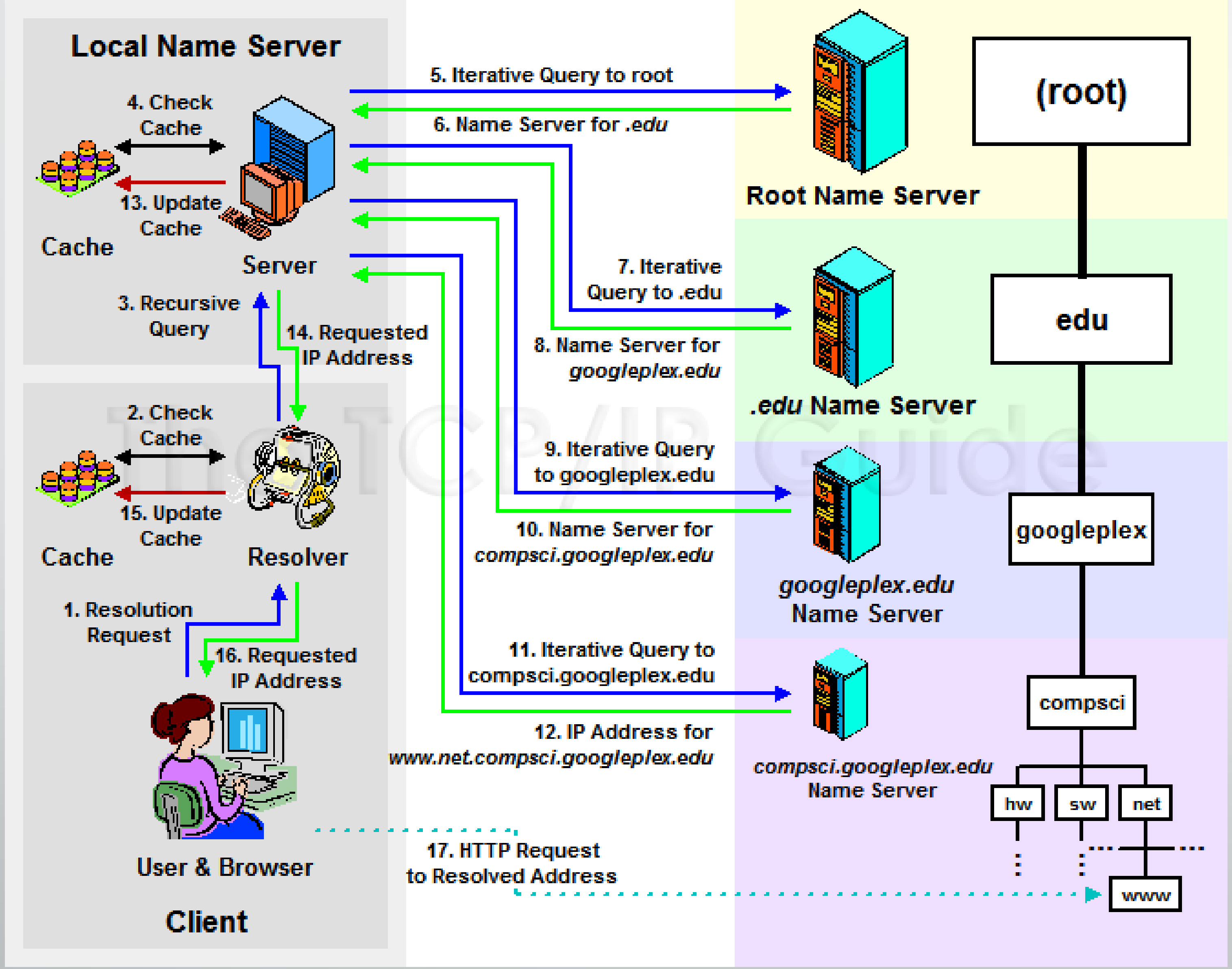
Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

Recursive query:

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



DNS Queries - Summary


Local Name Server
4. Check Cache
13. Update Cache
Cache
Server
3. Recursive Query
14. Requested IP Address
2. Check Cache
15. Update Cache
Cache
Resolver
1. Resolution Request
16. Requested IP Address
User & Browser
Client
5. Iterative Query to root
6. Name Server for .edu
7. Iterative Query to .edu
8. Name Server for googleplex.edu
9. Iterative Query to googleplex.edu
10. Name Server for compsci.googleplex.edu
11. Iterative Query to compsci.googleplex.edu
12. IP Address for www.net.compsci.googleplex.edu
17. HTTP Request to Resolved Address
Root Name Server
.edu Name Server
googleplex.edu Name Server
compsci.googleplex.edu Name Server
(root)
edu
googleplex
compsci
hw
sw
net
www
The TCP/IP Guide

Source :https://foxutech.com/what-is-dns-and-how-it-works/how-dns-works/

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
- Update/notify mechanisms proposed IETF standard
  - RFC 2136

DNS records

DNS: **Distributed database storing resource records (RR)**

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

type=A

- **Name** is hostname
- **Value** is IP address
- (relay1.bar.foo.com, IP, A)

type=CNAME

- **Name** is alias name for some "canonical" (the real) name
- **Value** is canonical name
- (ibm.com, server.backup.ibm.com, CNAME)

type=NS

- **Name** is domain
- **Value** is hostname of authoritative name server for this domain
- (fi.com, dns.fi.com, NS)

type=MX

- **Value** is name of mail server associated with **name**
- **(fi.com, mail.fi.com,MX)**

- **e.g. MX and NS can be use to map same domain to multiple server

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:

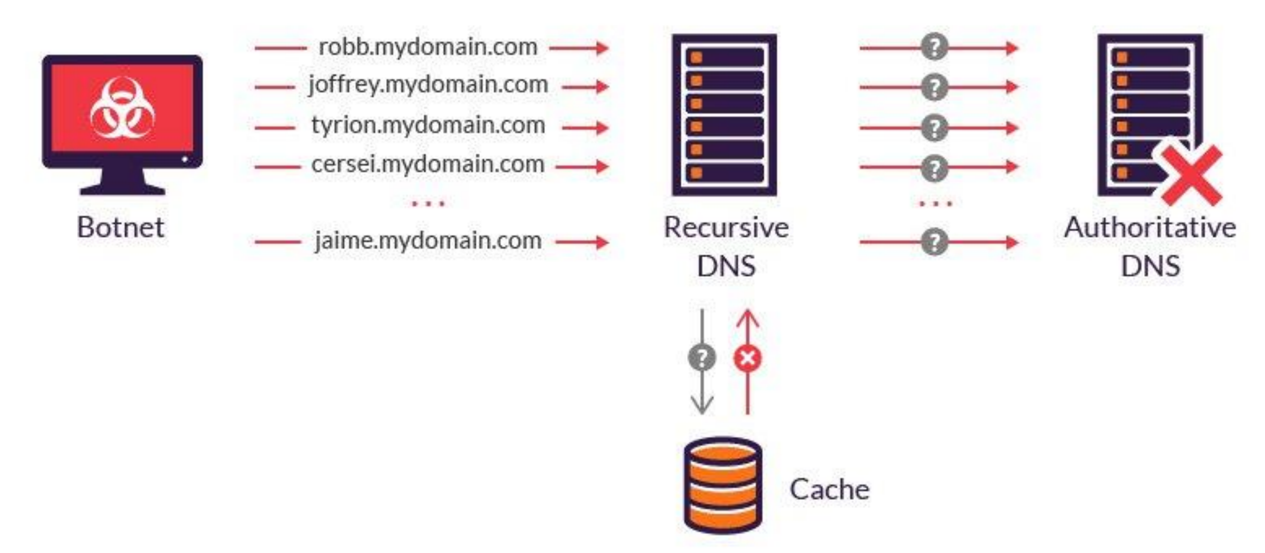
    ```
    (networkutopia.com, dns1.networkutopia.com, NS)

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

DNS security

DDoS attacks

- bombard root servers with traffic
  - not successful to date
  - traffic filtering
  - local DNS servers cache IPs of TLD servers, allowing root server bypass
- bombard TLD servers
  - potentially more dangerous



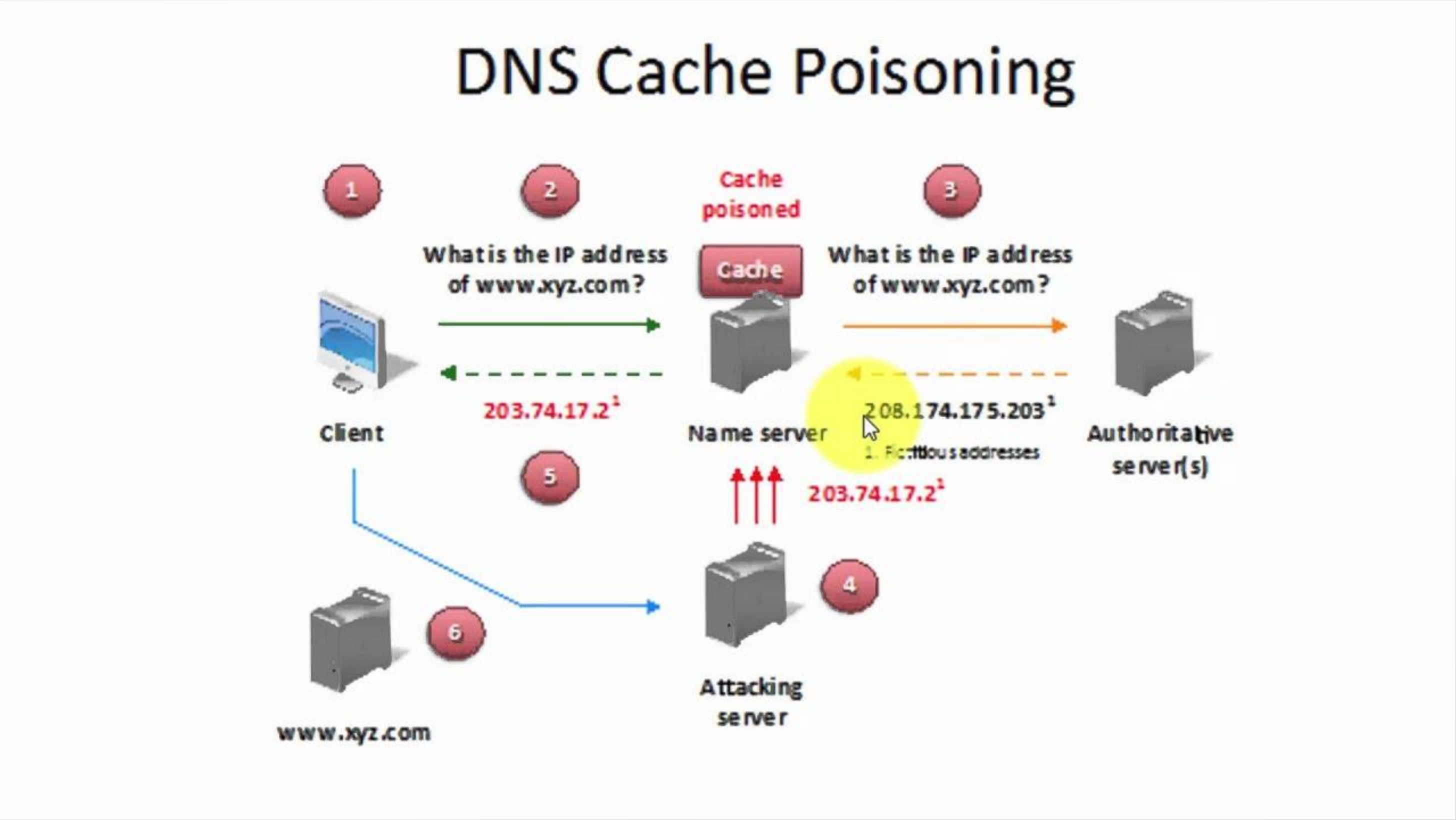
Source :https://www.imperva.com/learn/ddos/dns-flood/

DNS security

Spoofing attacks

- intercept DNS queries
- And then returns bogus replies
- Also known as

  DNS cache poisoning



Source : https://www.youtube.com/watch?v=71gpJ2wx7z8

- **Solution**
  - RFC 4033: DNSSEC authentication services
  - Uses public key cryptography (a way of digitally signing information) to verify and authenticate data.

THE END OF EMAIL AND DNS