Application Layer (Electronic Mail &DNS)

Lecture 3
CSE421 – Computer Networks

Department of Computer Science and Engineering
School of Data & Science

All material copyright 1996-2020
J.F Kurose and K.W. Ross, All Rights Reserved

Application Layer :Objectives

- Principles of network applications
- Web and HTTP
- **Electronic mail**
  - **SMTP, POP3, IMAP**
- DNS

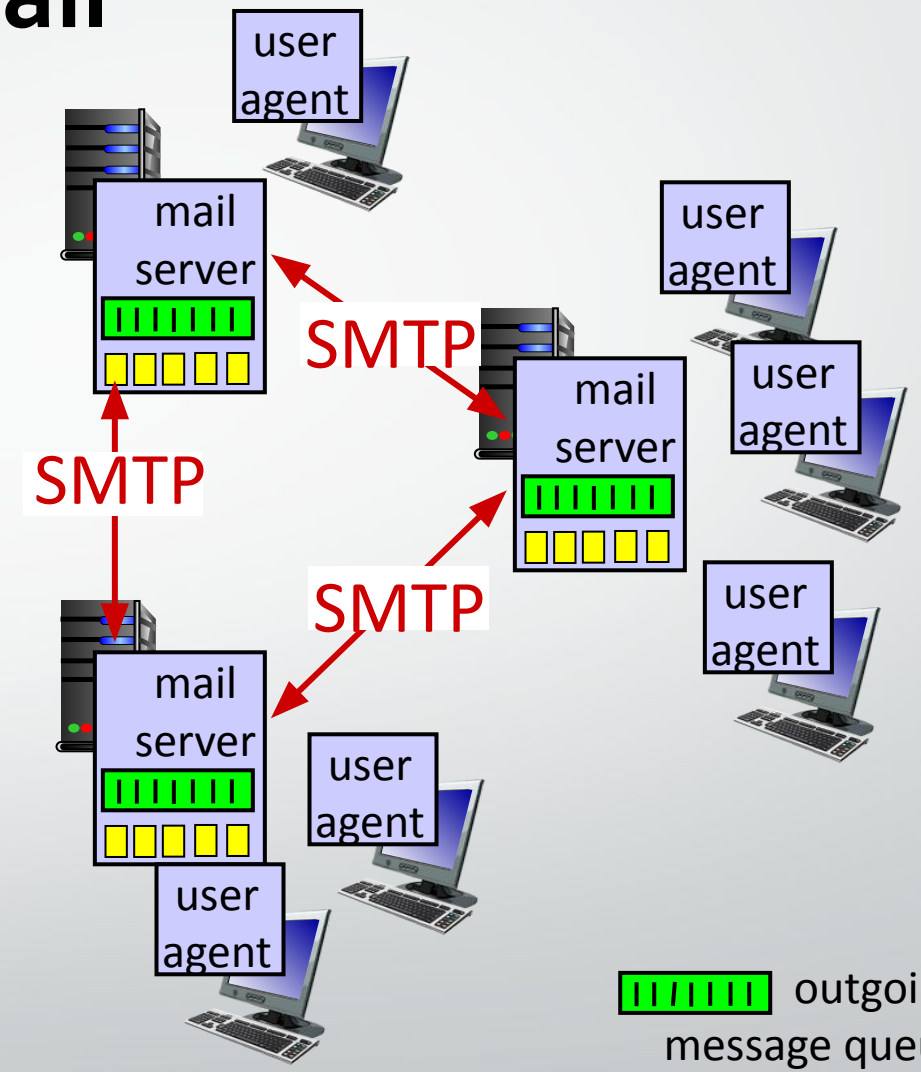
Electronic mail

Three major components:

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

User Agent

- Software program that is used for
- composing, editing, reading, forwarding mail messages
- e.g., Outlook, iPhone mail client, Web browser



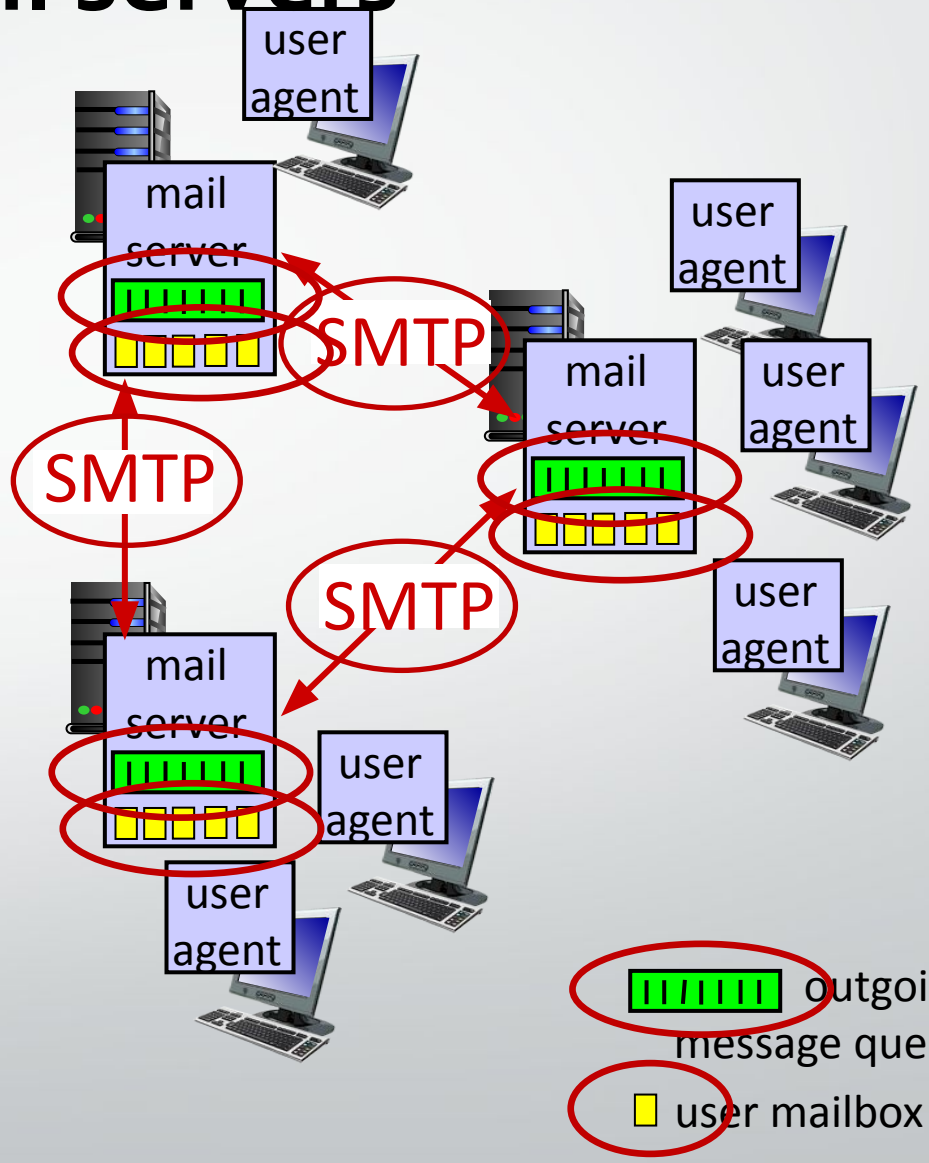
Electronic mail: mail servers

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



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

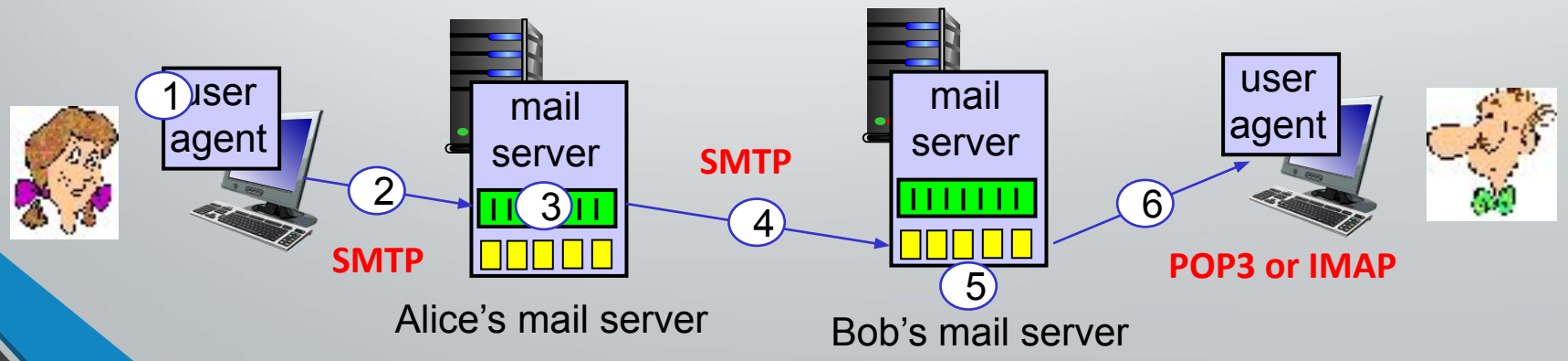
3) client side of SMTP at mail server opens TCP connection with Bob's mail server

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

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

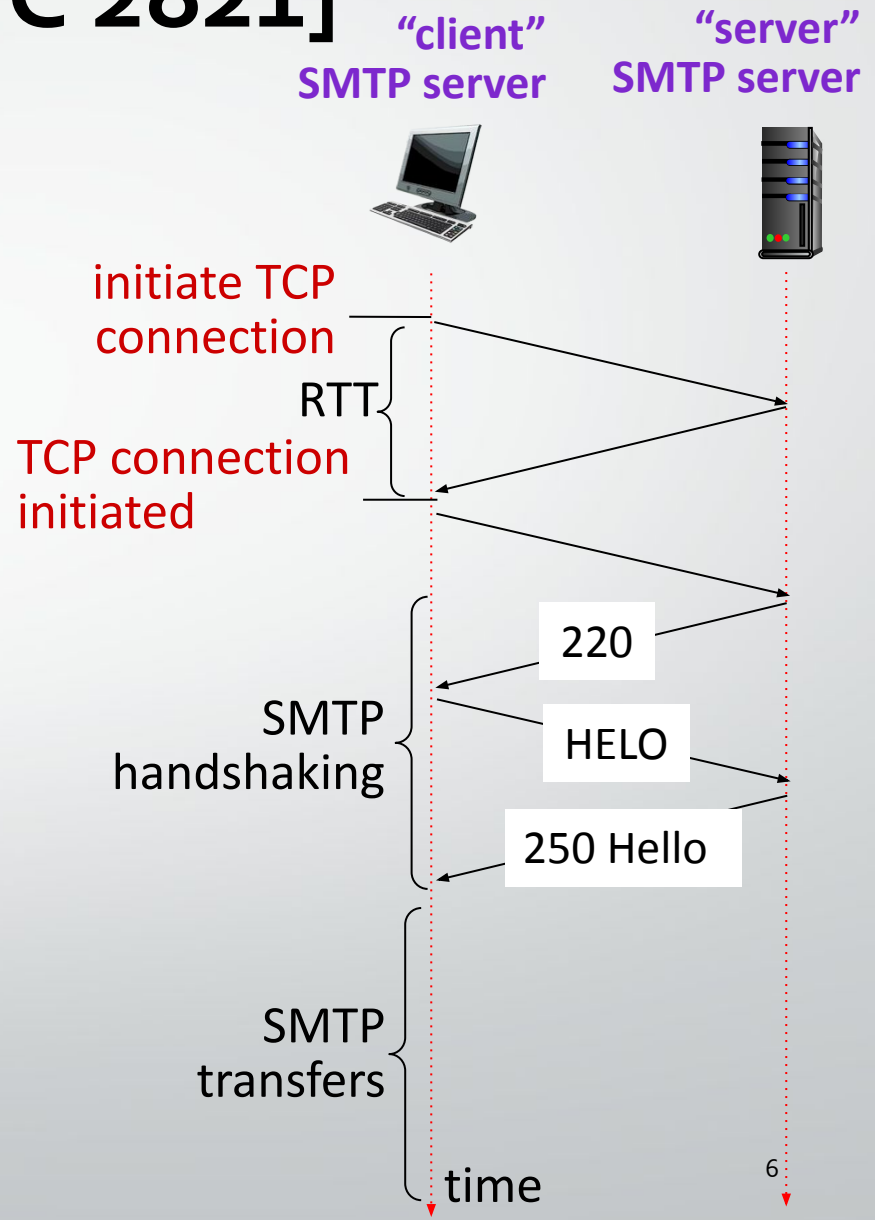
6) Bob invokes his user agent to read message using **POP3 or IMAP.**

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



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



Sample SMTP interaction

SMTP handshaking

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
```

SMTP header

SMTP Message Data Transfer

SMTP Termination

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

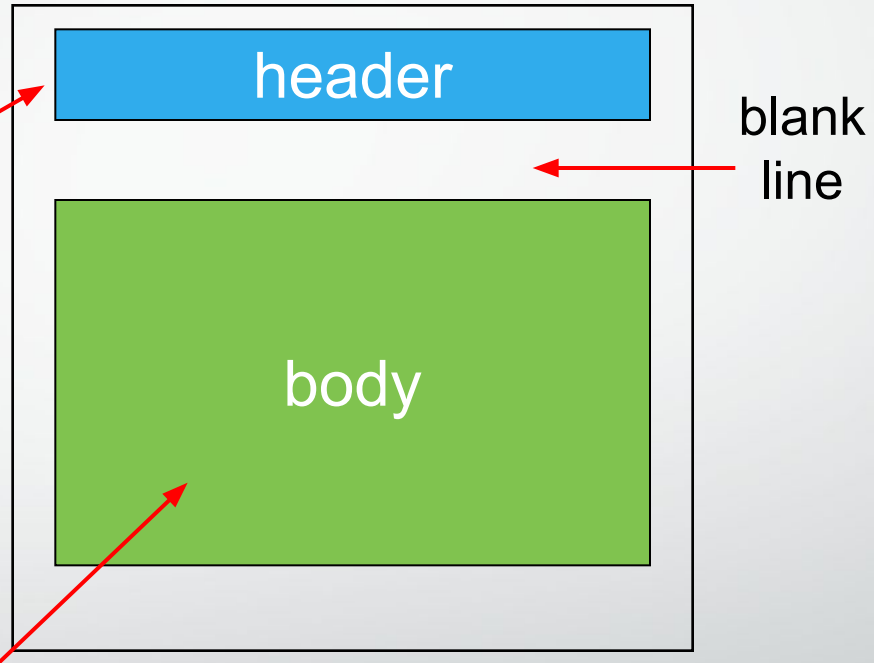
RFC 822: standard for text message format:

- header lines, e.g.,

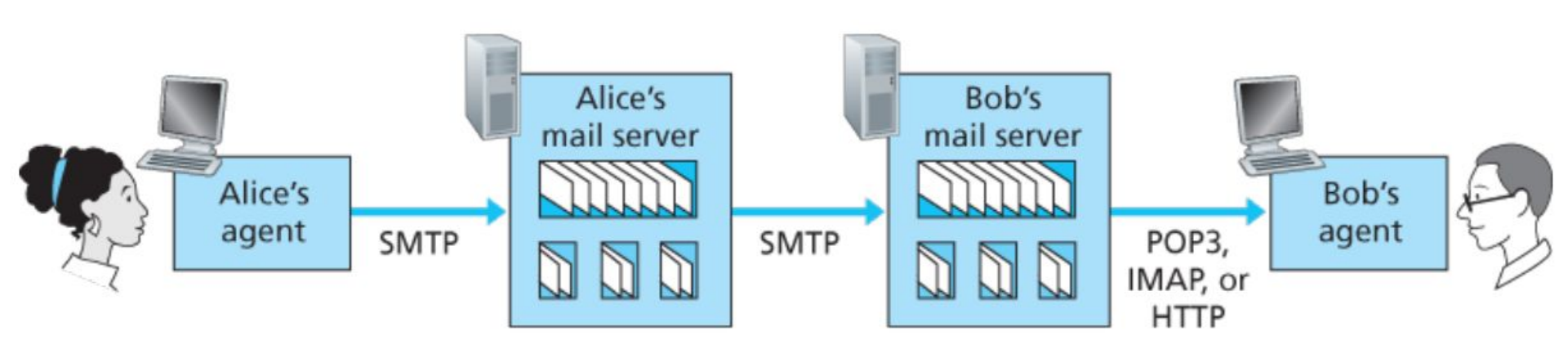
```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Searching for the
```

different from SMTP MAIL FROM, RCPT TO: commands!

- Body: the "message"
  - ASCII characters only



Mail access protocols



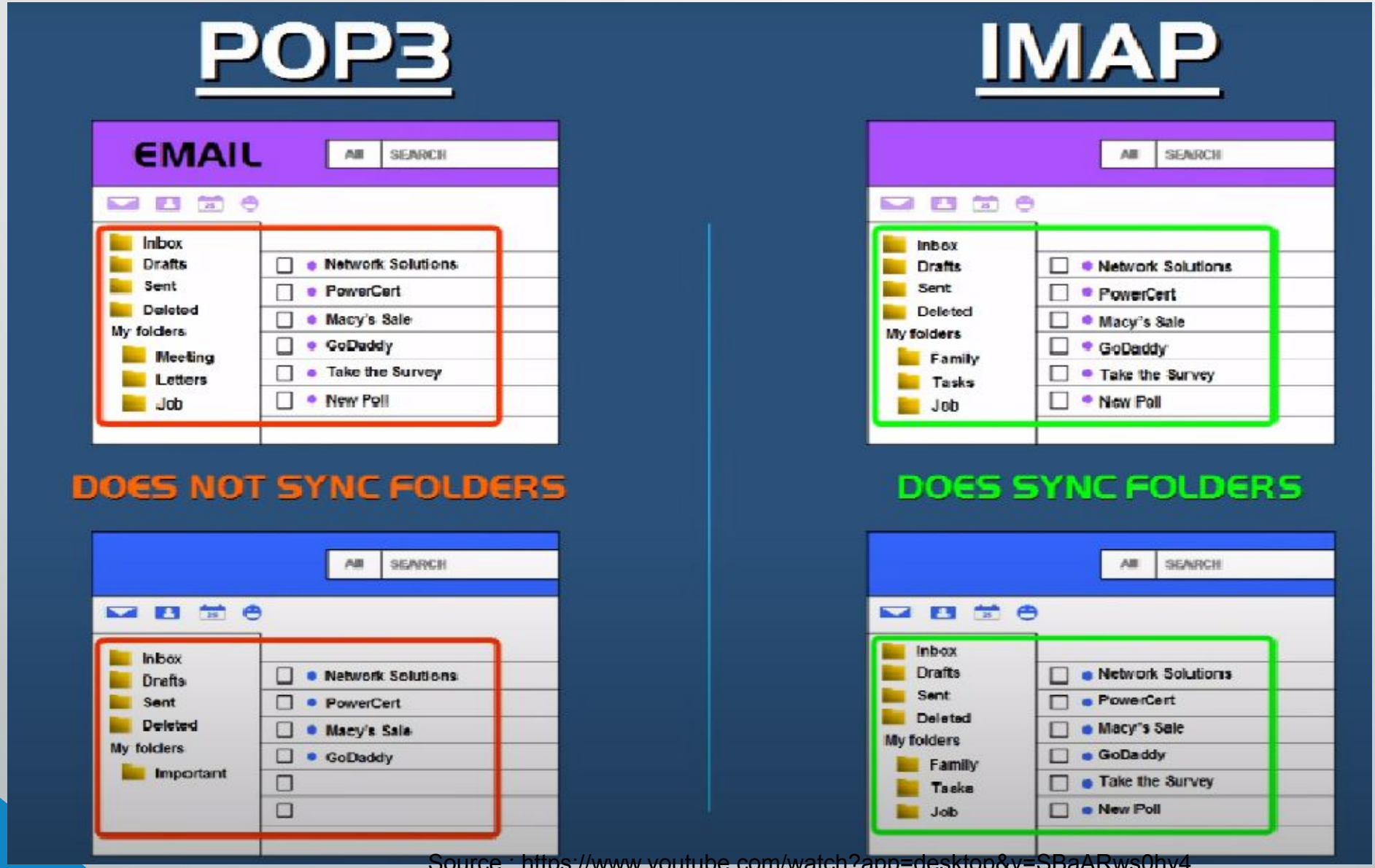
- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]: authorization, download
  - IMAP: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored messages on server
  - HTTPs: Web based(Gmail,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

POP3 vs 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 vs IMAP



Source : https://www.youtube.com/watch?app=desktop&v=SBaARws0hy4

Application Layer :Objectives

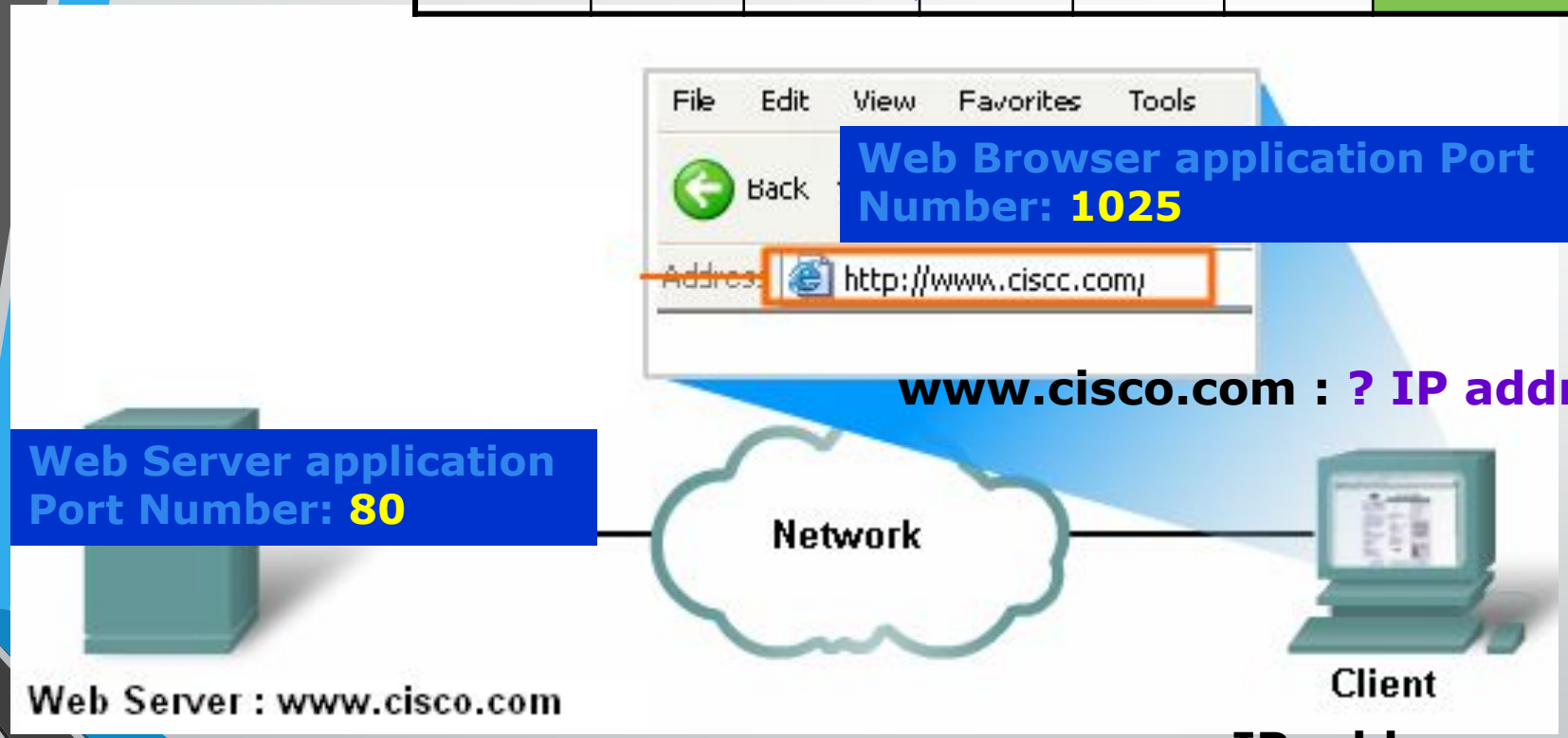
- Principles of network applications
- Web and HTTP
- Electronic mail
  - SMTP, POP3, IMAP
- **DNS**

Domain Name System (DNS)

- DNS is the phone book of the Internet.

Domain Name System (DNS)

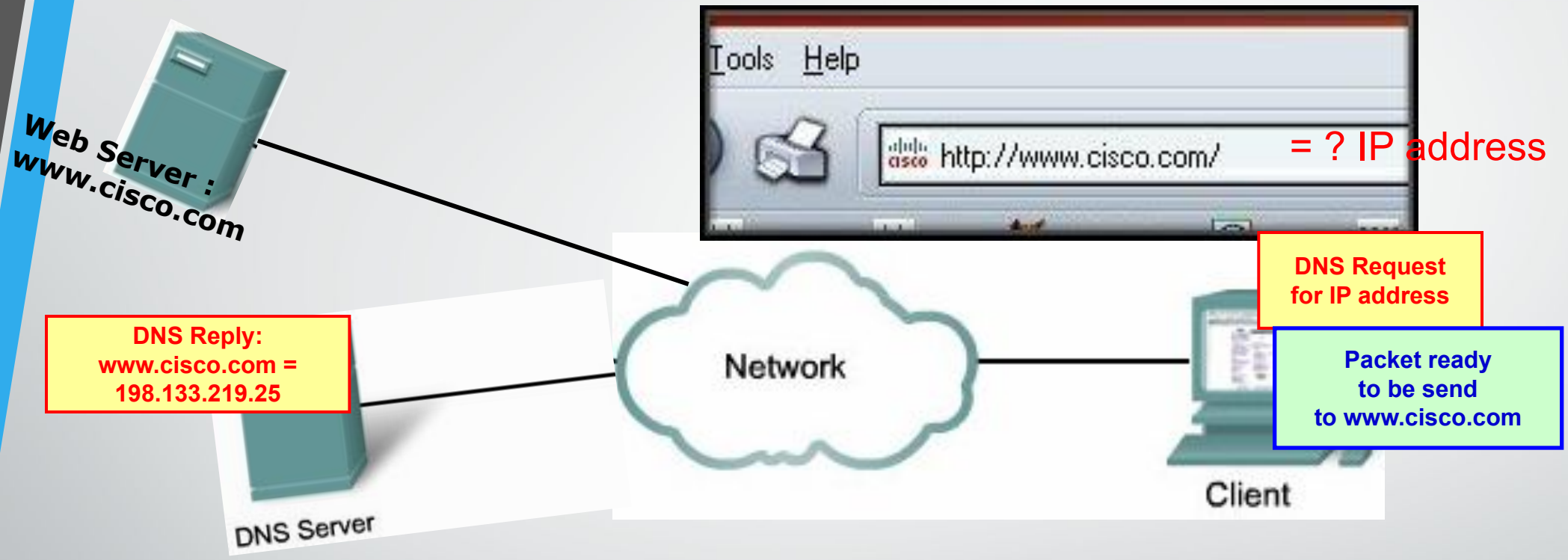
Dest MAC Address	Source MAC Address	Destination IP Address	Source IP Address	Destination Port No	Source Port No		
		?	192.168.1.1	80	1025	**Data: Request for web page**	Trailer



IP address :
200.20.20.1

IP address :
192.168.1.1

Domain Name System (DNS)



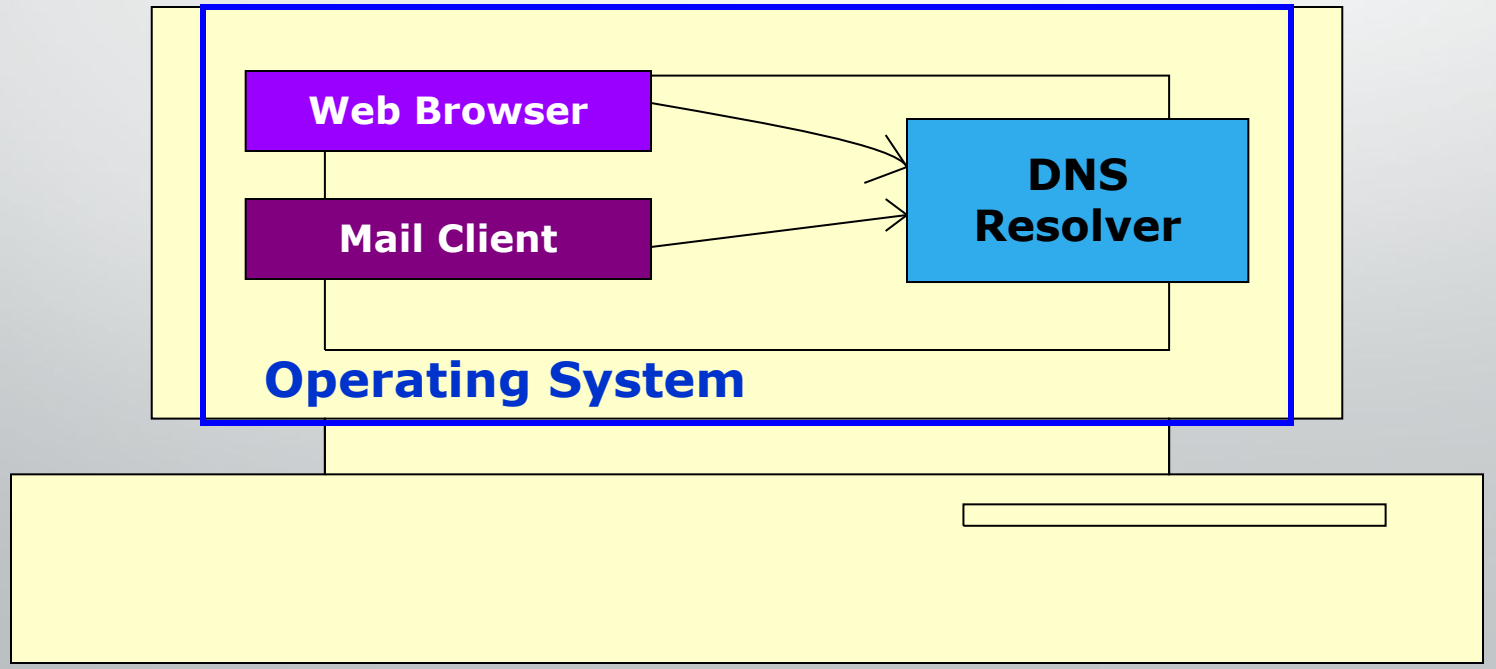
DNS Address Book:

www.cisco.com = 198.133.219.25

www.yahoo.com = 200.133.2.56

Domain Name System (DNS)

- DNS is an **automated client/server** service.
- Internet programs requiring domain name look up send a resolution request to the **DNS resolver** (Client side of DNS) in the local operating system
- The resolver in turn handles the communications required.



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

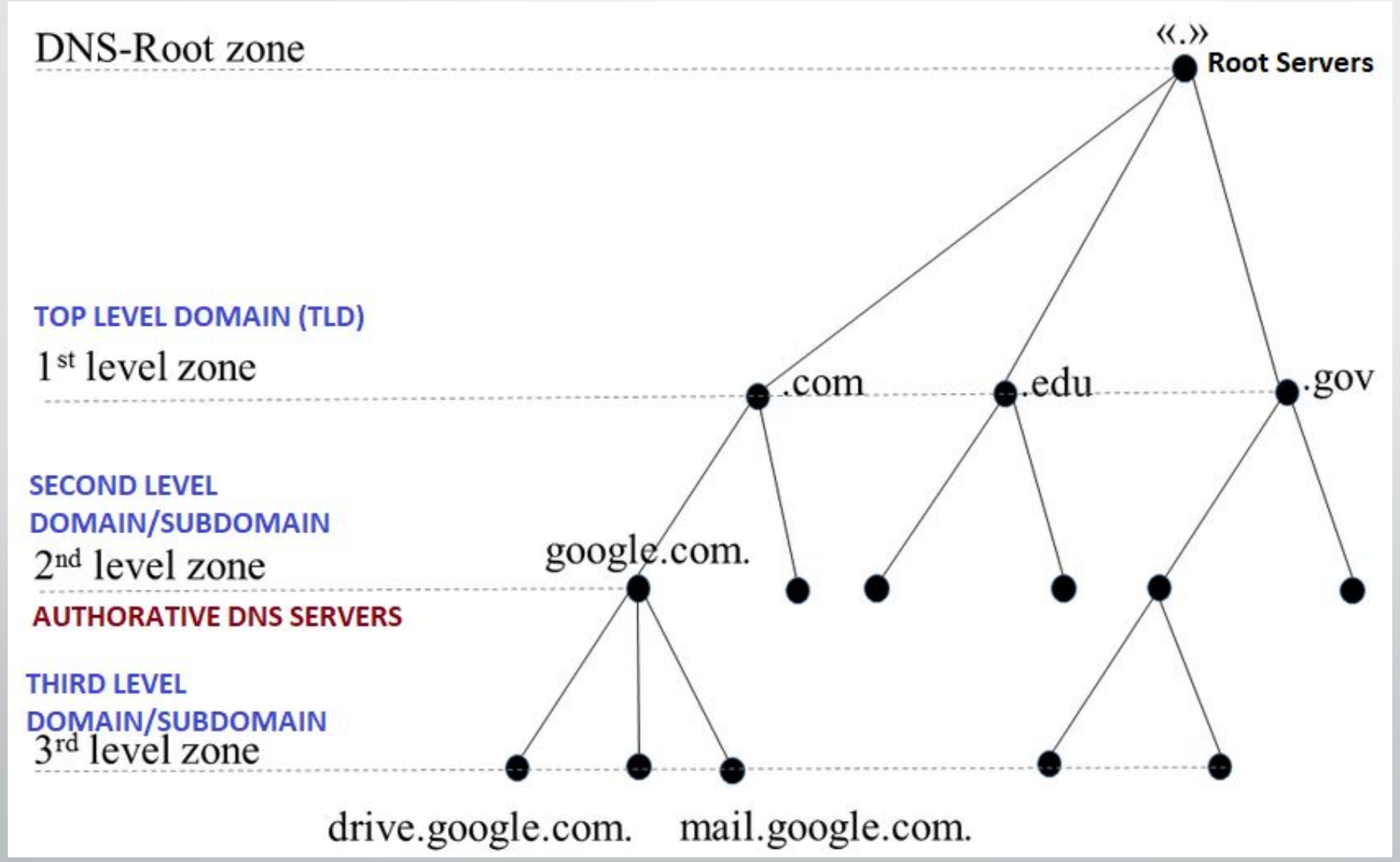
DNS Name Servers

Centralized DNS? **NO**

REASONS ?

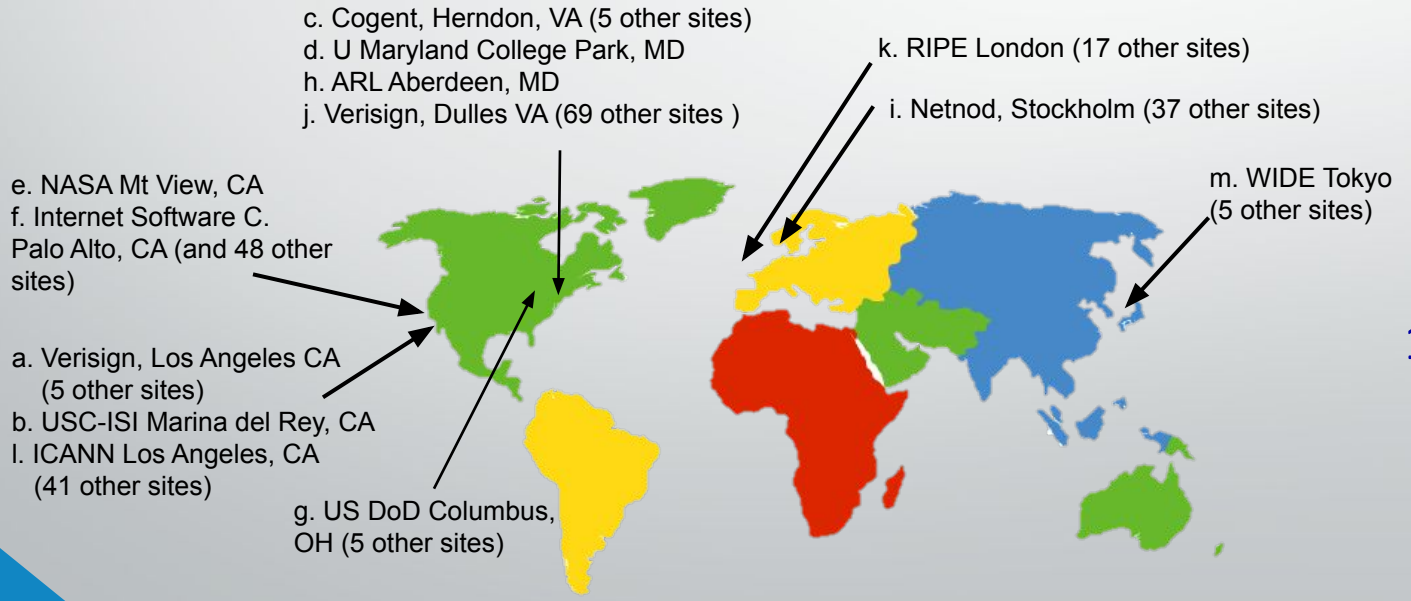
- Single point of failure
- Traffic volume
- Distance centralized database
- Maintenance
- Doesn't *scale!*
- *Solution: Distributed Database*

DNS: a distributed, hierarchical database



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!
- ICANN (Internet Corporation for Assigned Names and Numbers) manages root DNS domain

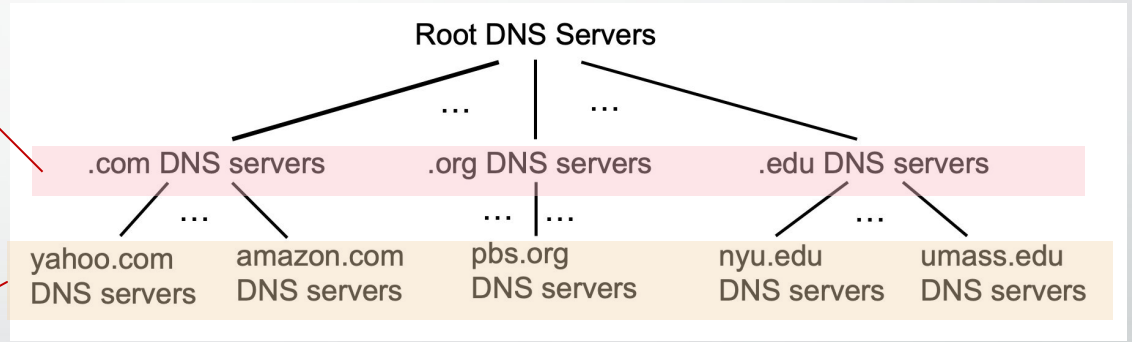


13 logical root name "servers" worldwide each "server" replicated many times (~200 servers in US)

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

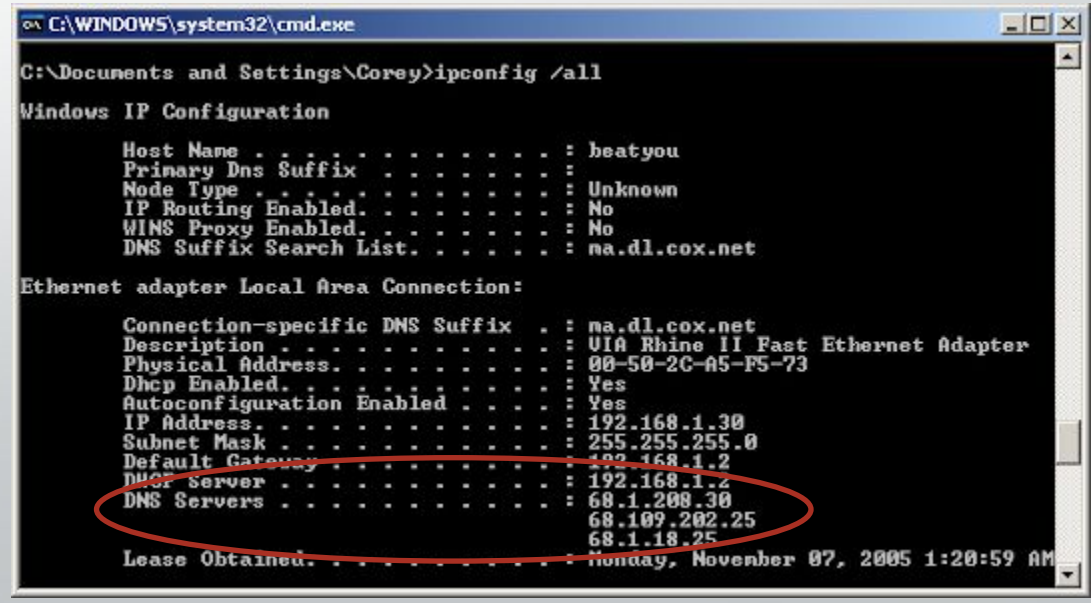


Second Level Domain/Authoritative DNS servers:

- 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

- 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`

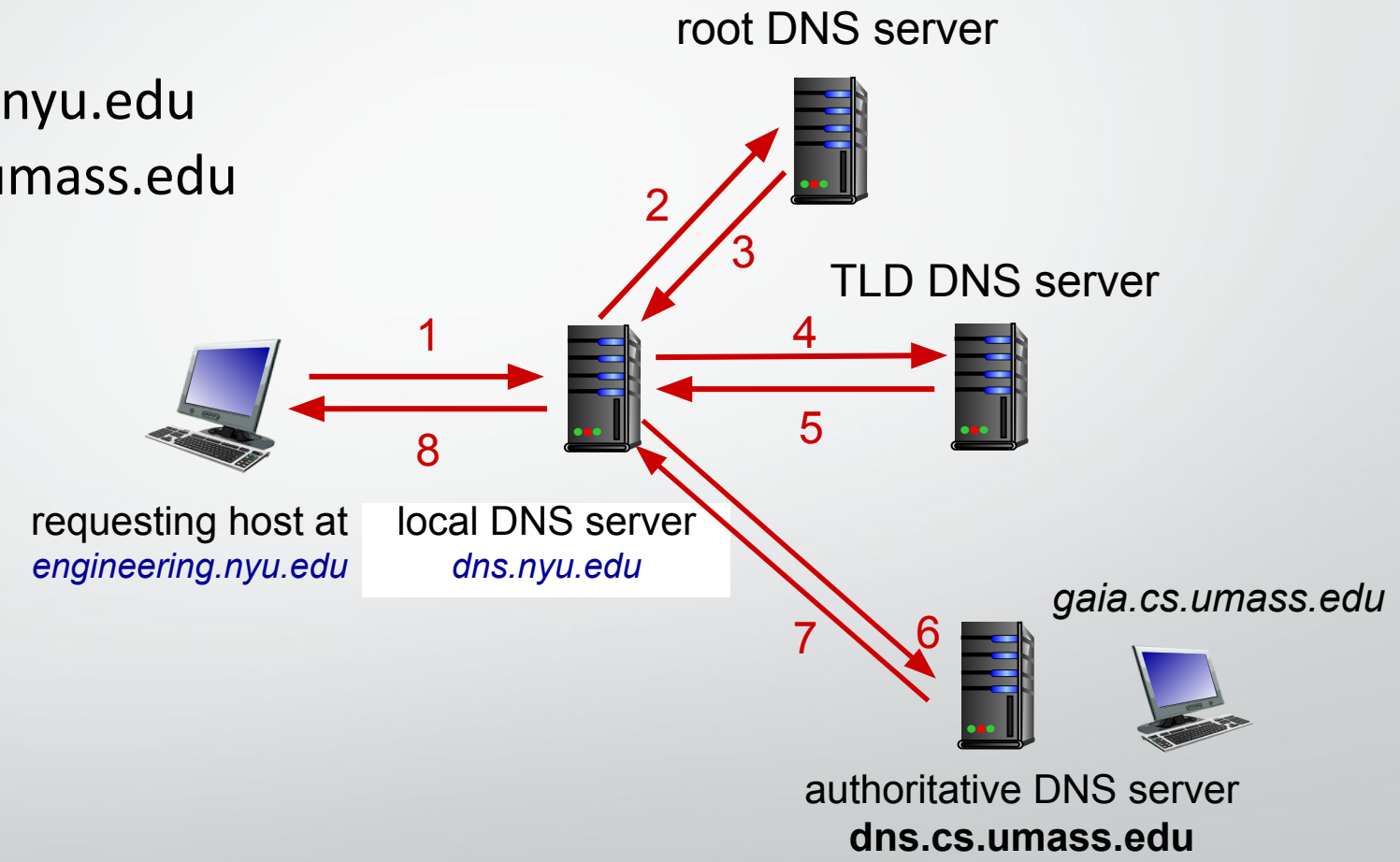


DNS name resolution: iterated query

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"

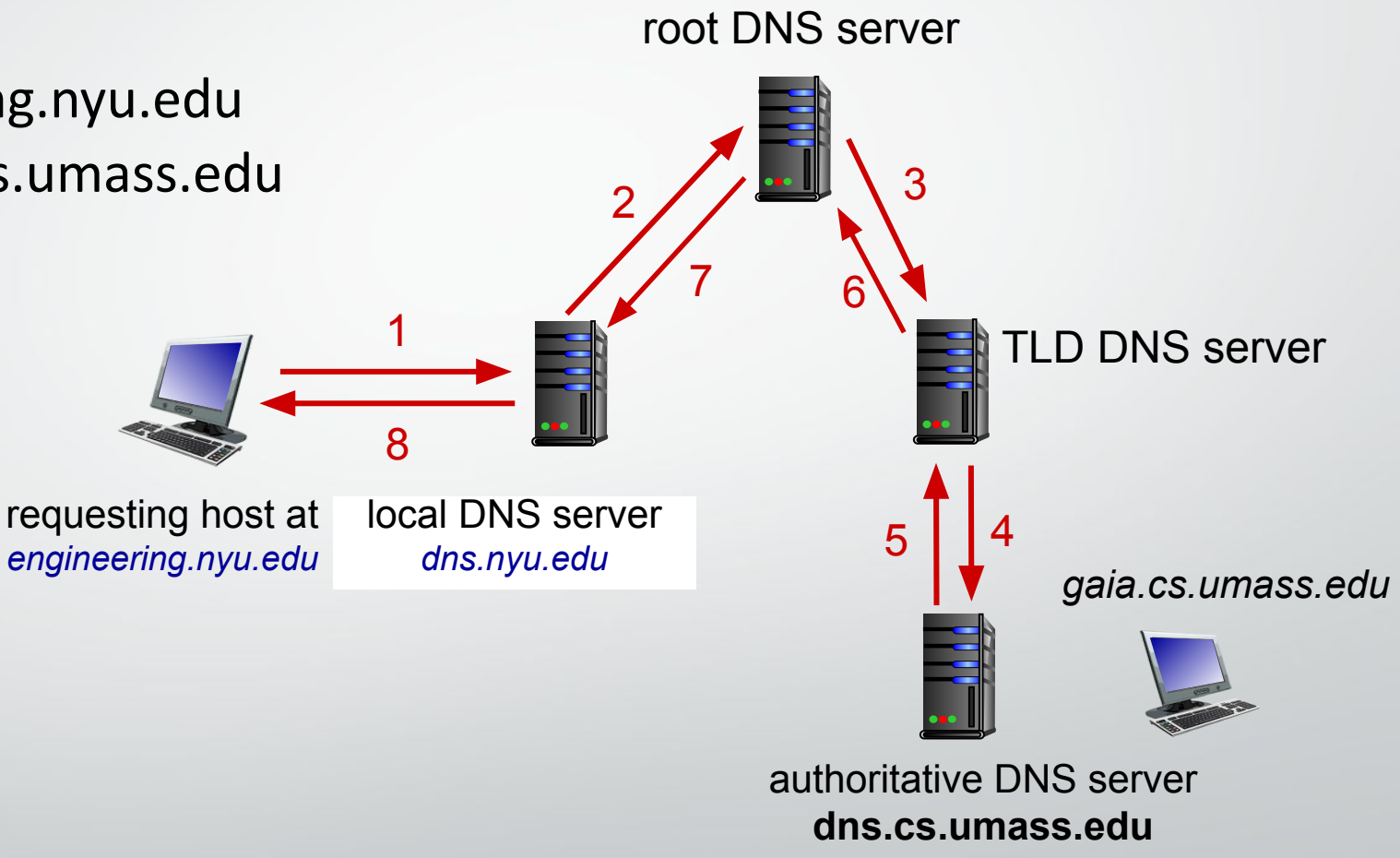


DNS name resolution: recursive query

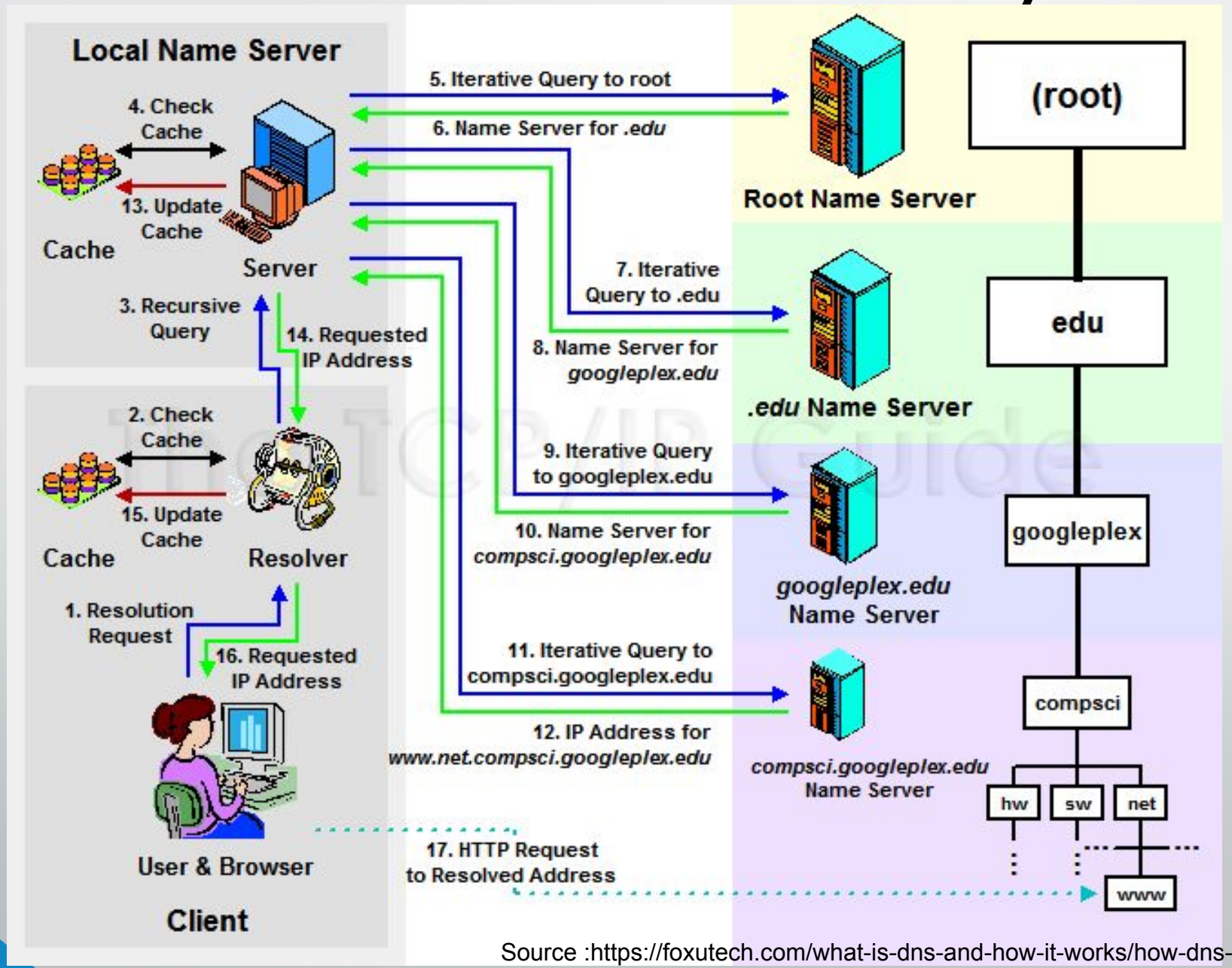
Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

Recursive query:

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



DNS Queries - Summary



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

DNS: caching, updating records

- **Caching**: When a name server learns an IP address for a domain, it saves (caches) this information for faster access next time.
- **TTL (Time to Live)**: Cached information has a lifespan (TTL) and removed from the cache after this time, requiring a fresh lookup.
- **Local Caching**: Local name servers often cache TLD (Top-Level Domain) records to reduce the load on root servers.
- **Out-of-Date Cache**: Cached entries might be outdated if a domain's IP address changes. The change only becomes known across the internet once all caches with the old IP expire.
- **Updating Standard**: There's an IETF standard (RFC 2136) for updating DNS records to address this issue, allowing quicker updates across servers.

DNS Records

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

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

type=A

- **Name** is hostname
- **Value** is IP address
- (google.com, 172.10.12.32, A)

type=NS

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

type=CNAME

- **Name** is alias name for some "canonical" (the real) name
- **Value** is canonical name
- (google.com, www.google.com, CNAME)
- (mail.google.com, google.com, CNAME)

type=MX

- **Value** is name of mail server associated with **name**
- (google.com, mail.google.com,MX)
- (mail.google.com,172.10.12.39, A)

Inserting records into DNS

- Example: new startup "**Network Utopia**"
- Register name networkuptopia.com at ***DNS registrar*** (e.g., Network Solutions)
- Network Utopia : Web Server

  (networkutopia.com, 212.212.71.4, A)

  - **CNAME** :

  (networkutopia.com, www.networkutopia.com, CNAME)

  - Registrar inserts a RR into .com TLD server:
    - Local Primary DNS Server

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

    - Email Server

    (networkutopia.com, mail.networkutopia.com, MX)

    (mail.networkutopia.com, 212.212.73.6, A)

THE END OF EMAIL AND DNS