CSCI 466: Networks

Application Layer

(More DNS, SMTP)

Reese Pearsall

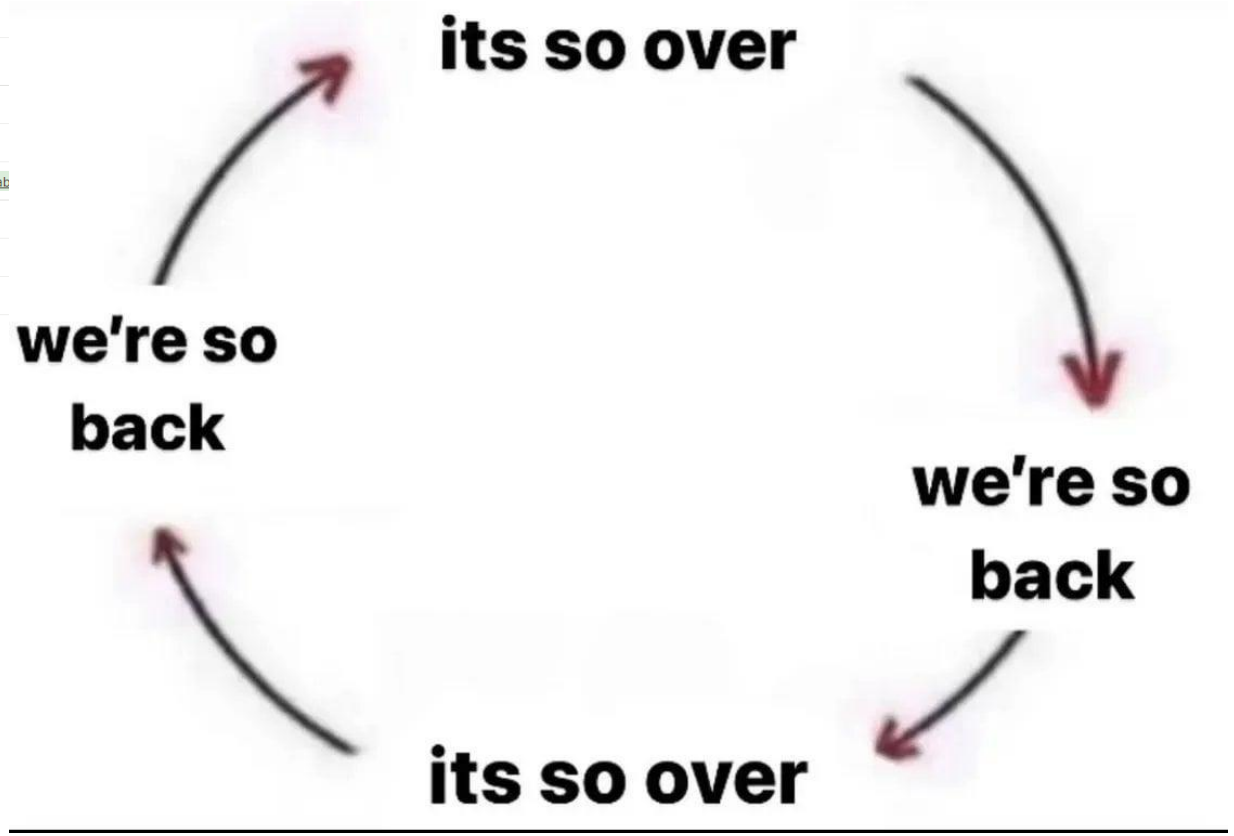
Fall 2024

*All images are stolen from the internet

CSCI 466: Networks 📶

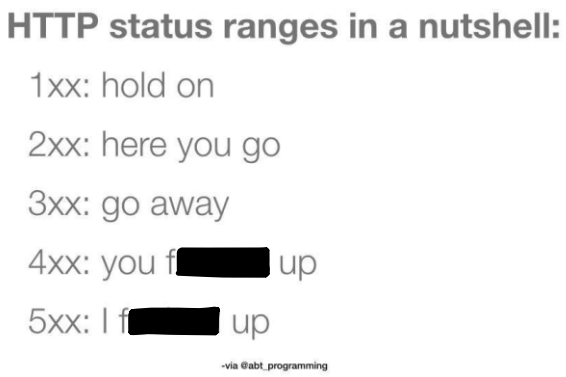
Fall 2024

Date	Topic	Extra Notes	Slides + Lecture Recordings		Assignment
Wednesday August 21st	Syllabus		Slides	Lecture Recording	Please Fill out the Course Questionnaire!
Friday August 23rd	Internet Structure, Data Forwarding		Slides	Lecture Recording	
Monday August 26th	OSI Model, Data Forwarding		Slides	Lecture Recording	
Wednesday August 28th	Network Performance		Slides	Lecture Recording	
Friday August 30th	Application Layer + HTTP		Slides	Lecture Recording	
Monday September 2nd	OFF NO CLASS				
Wednesday September 4th	HTTP Requests, Wireshark		Slides	Lecture Recording	
Friday September 6th	(Asyncronous class) Git, Socket Programming		Code	Lecture Recording	Quiz 1
Monday September 9th	DNS		Slides	Lecture Recording	
Wednesday September 11th	DNS, SMTP				
Friday September 13th	FTP, P2P, CDNs				Wireshark Lab
Monday September 16th	Transport Layer				
Wednesday September 18th	Transport Layer				
Friday September 20th	PA1 + Quiz 2 Work Day (No lecture)				Quiz 2



Course Website is back after DNS was updated☺

Announcements



Wireshark Lab 1 due on Friday @ 11:59 PM

Cybersecurity Capture the Flag (CTF) club meets on Fridays

- Open to anyone
- We will be competing in an online CTF event next weekend

https://learn2ctf.org/

The system of rules that define the format for how devices should communicate on the internet is known as what?

- Access Network
- Physical Medium
- The OSI Model
- A protocol

What is the purpose of the **network core**?

- To connect networks to other networks
- To route and transfer data to the correct destination
- To connect ISP to other ISPs
- All of the above

Finding the TCP Handshake in Wireshark

ip.addr == 128.119.245.12

No.	Time	Source	Destination	Protocol	Length	Info
67	2.225618	153.90.118.85	128.119.245.12	TCP	66	49809 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
85	2.292657	128.119.245.12	153.90.118.85	TCP	66	80 → 49809 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1380 SACK_PERM=1 WS=128
86	2.292723	153.90.118.85	128.119.245.12	TCP	54	49809 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
87	2.292847	153.90.118.85	128.119.245.12	HTTP	175	GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
109	2.361804	128.119.245.12	153.90.118.85	TCP	60	80 → 49809 [ACK] Seq=1 Ack=122 Win=29312 Len=0
110	2.363476	128.119.245.12	153.90.118.85	HTTP	436	HTTP/1.1 200 OK (text/html)
111	2.364246	153.90.118.85	128.119.245.12	TCP	54	49809 → 80 [FIN, ACK] Seq=122 Ack=383 Win=261632 Len=0
113	2.431143	128.119.245.12	153.90.118.85	TCP	60	80 → 49809 [FIN, ACK] Seq=383 Ack=123 Win=29312 Len=0
114	2.431217	153.90.118.85	128.119.245.12	TCP	54	49809 → 80 [ACK] Seq=123 Ack=384 Win=261632 Len=0

OSI Model

Application Layer

Presentation Layer *

Session Layer *

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Application Layer

Messages from Network Applications

↓

Physical Layer

Bits being transmitted over a copper wire

**In the textbook, they condense it to a 5-layer model, but 7 layers is what is most used*

DNS

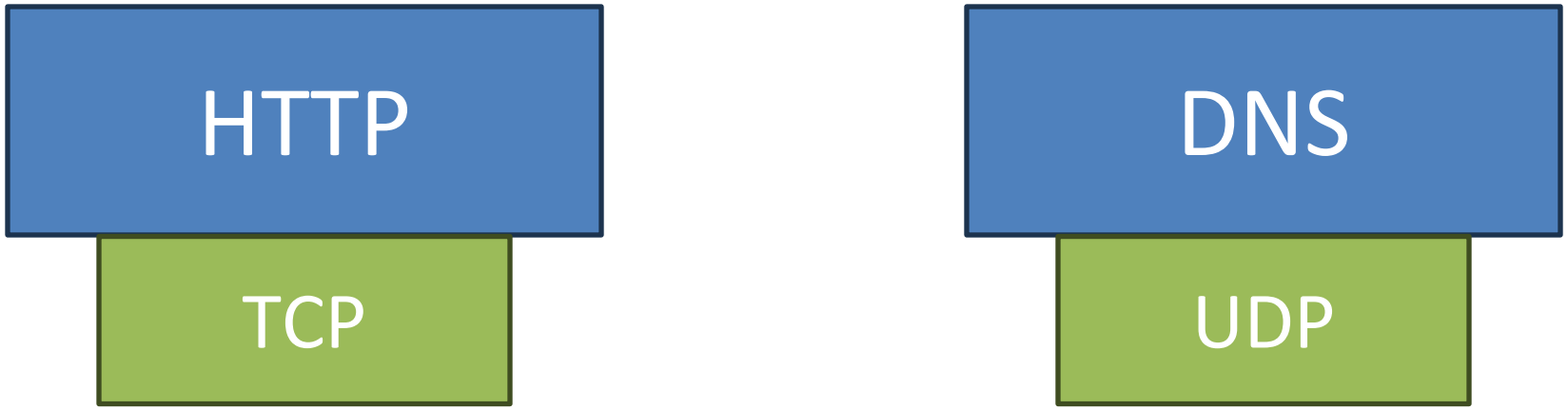
Humans browse the web using hostnames
- (They need English)

Computers understand numbers
- (They need IP addresses)

→ **DNS** → `153.90.127.197`

Domain Name System (DNS) is a database of mappings between hostnames and IP addresses

Protocols so far



DNS Architecture

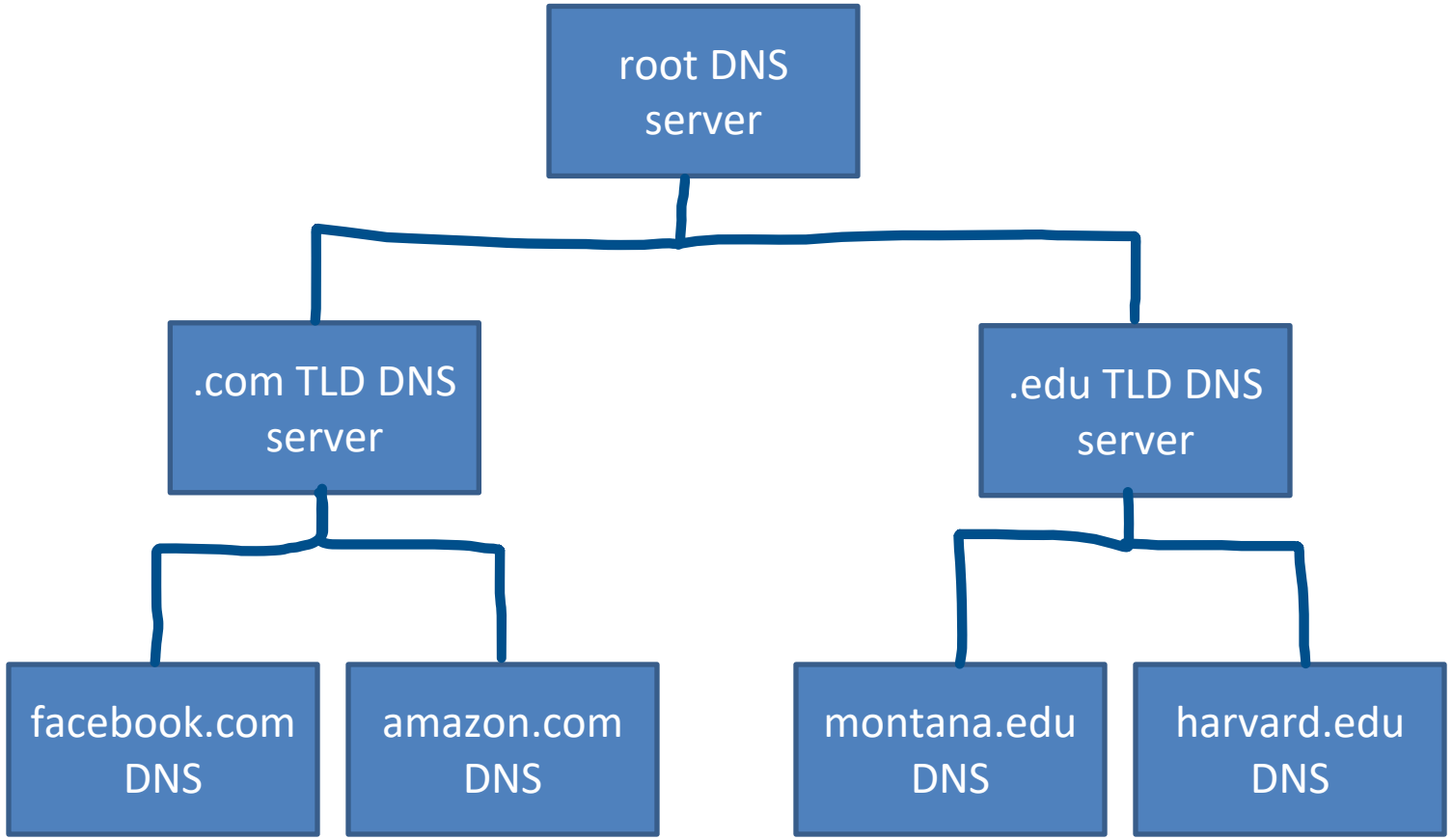
- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hierarchy consists of different types of DNS servers:

Authoritative DNS servers- Organization's own DNS with up-to-date records

Top-level domain (TLD) servers- responsible for keeping IP addresses for authoritative DNS servers for each top-level domain (.com, .edu, .jp, etc)

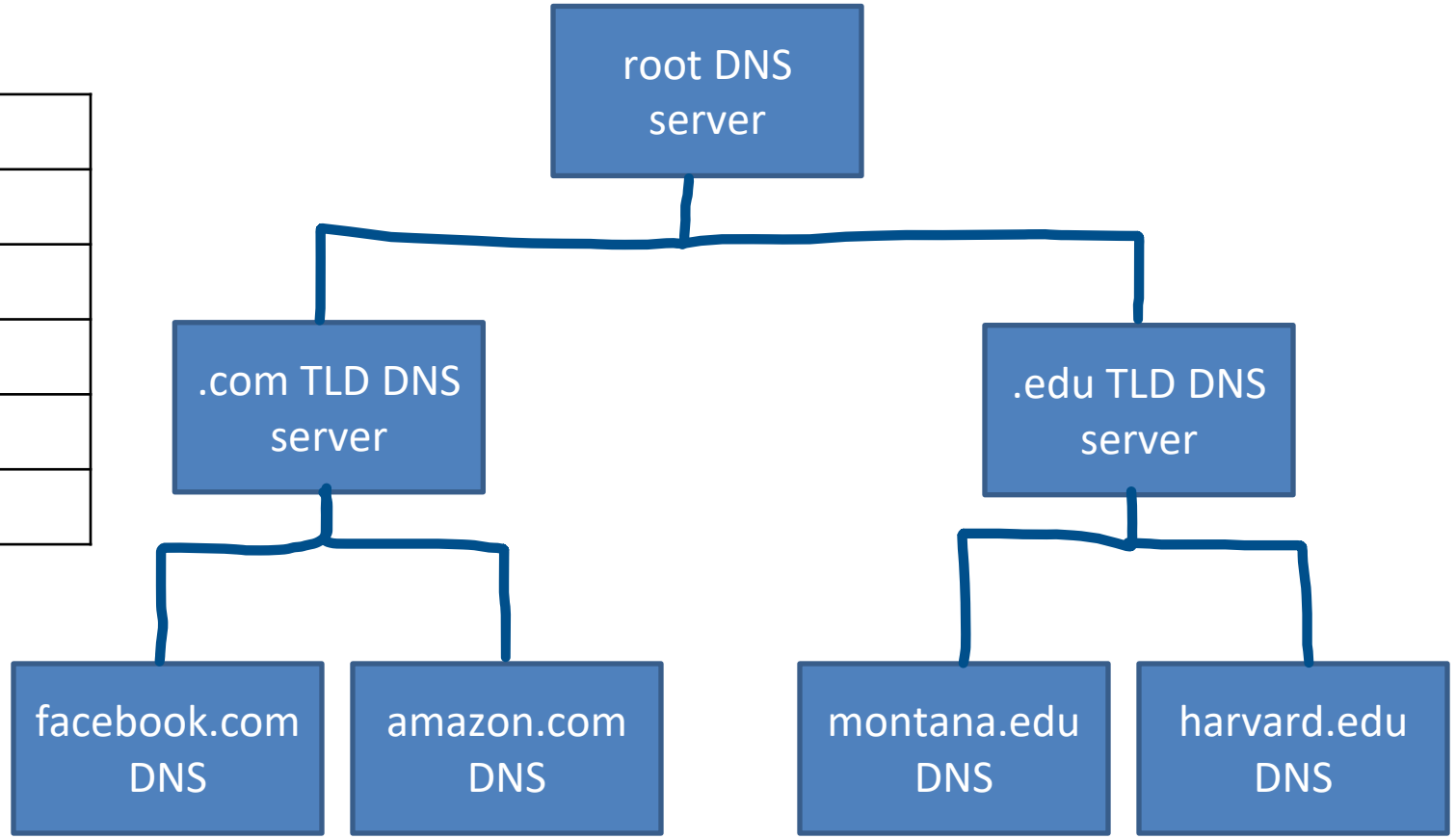
Root DNS servers- responsible for maintaining IP addresses for TLD servers



DNS Architecture

- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hostname	IP Address
marketplace.facebook.com	192.23.54.221
gaming.facebook.com	192.23.54.219
facebook.com	192.23.54.222
friends.facebook.com	192.23.54.216
…	…

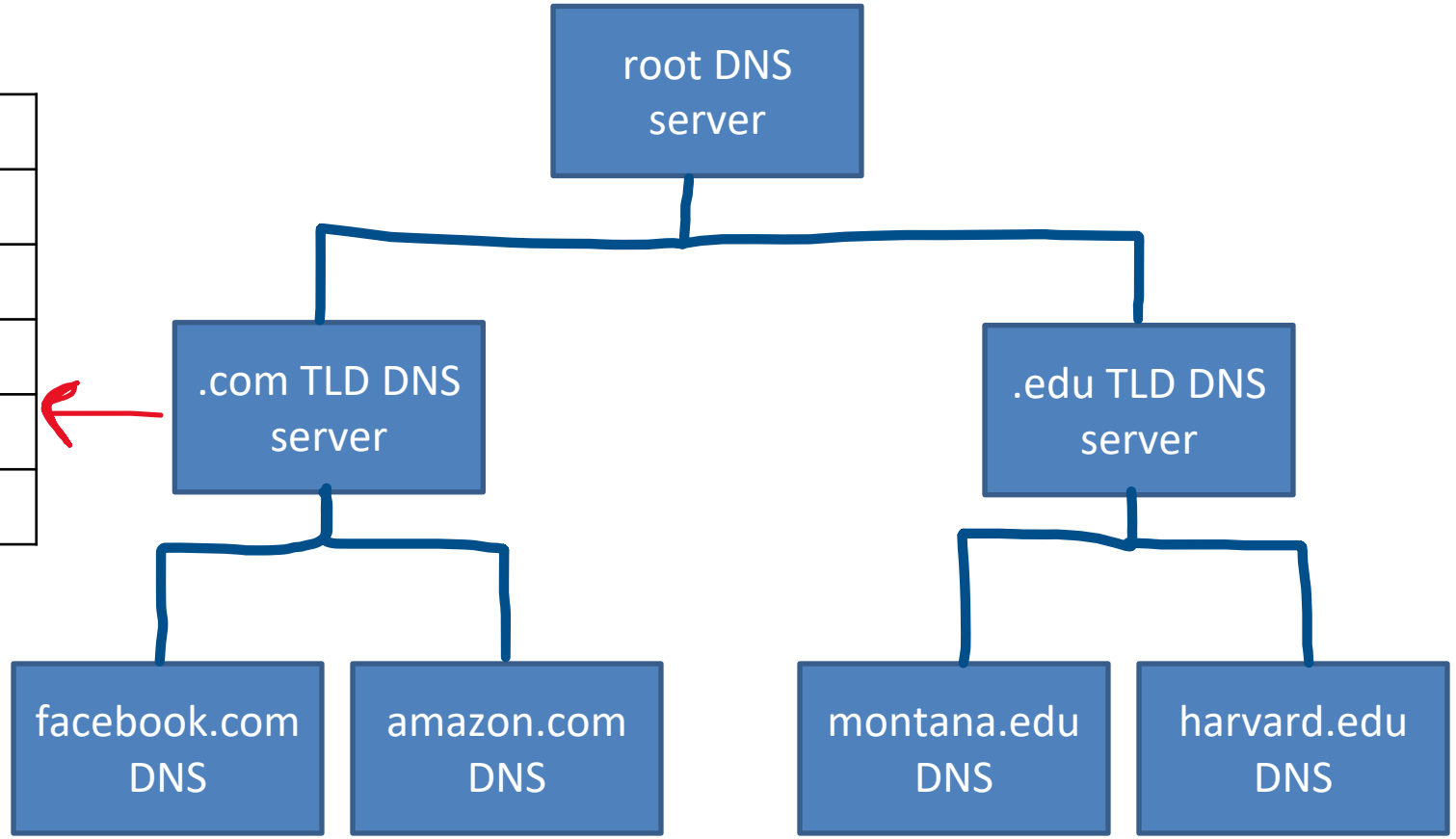


Authoritative DNS Server

DNS Architecture

- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hostname	IP Address
google**.com** Auth. DNS	77.87.124.3
facebook**.com** Auth. DNS	192.23.54.22
amazon**.com** Auth DNS	10.172.44.92
ebay**.com** Auth DNS	192.7.66.111
…	…

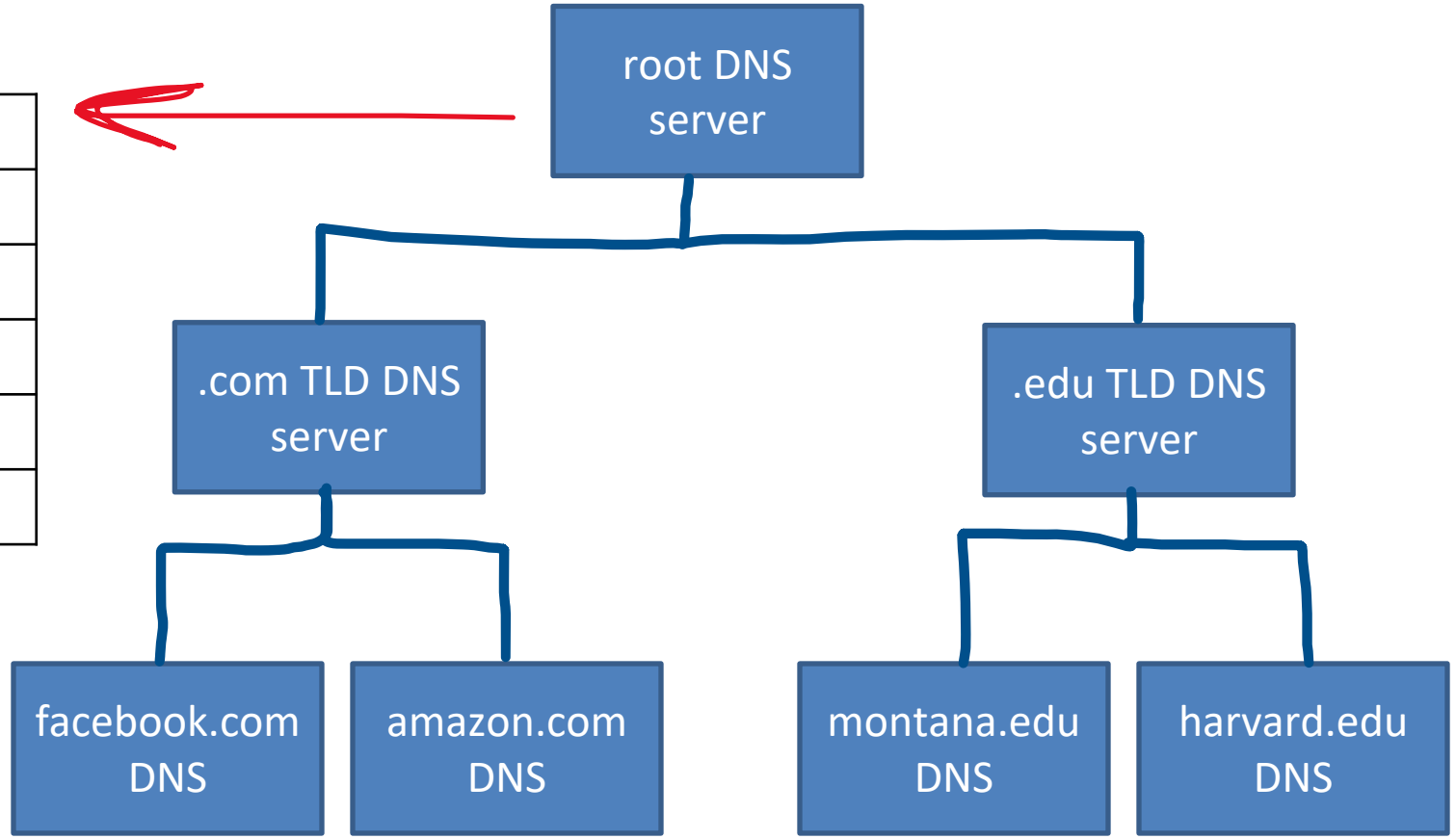


TLD DNS servers hold records for authoritative DNS server for a particular domain

DNS Architecture

- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

Hostname	IP Address
.com TLD DNS server	21.220.198.29
.org TLD DNS server	68.198.64.235
.edu TLD DNS server	103.109.123.65
.gov TLD DNS server	39.61.129.155
…	…



The root DNS server holds records for TLD DNS servers for all top-level domains

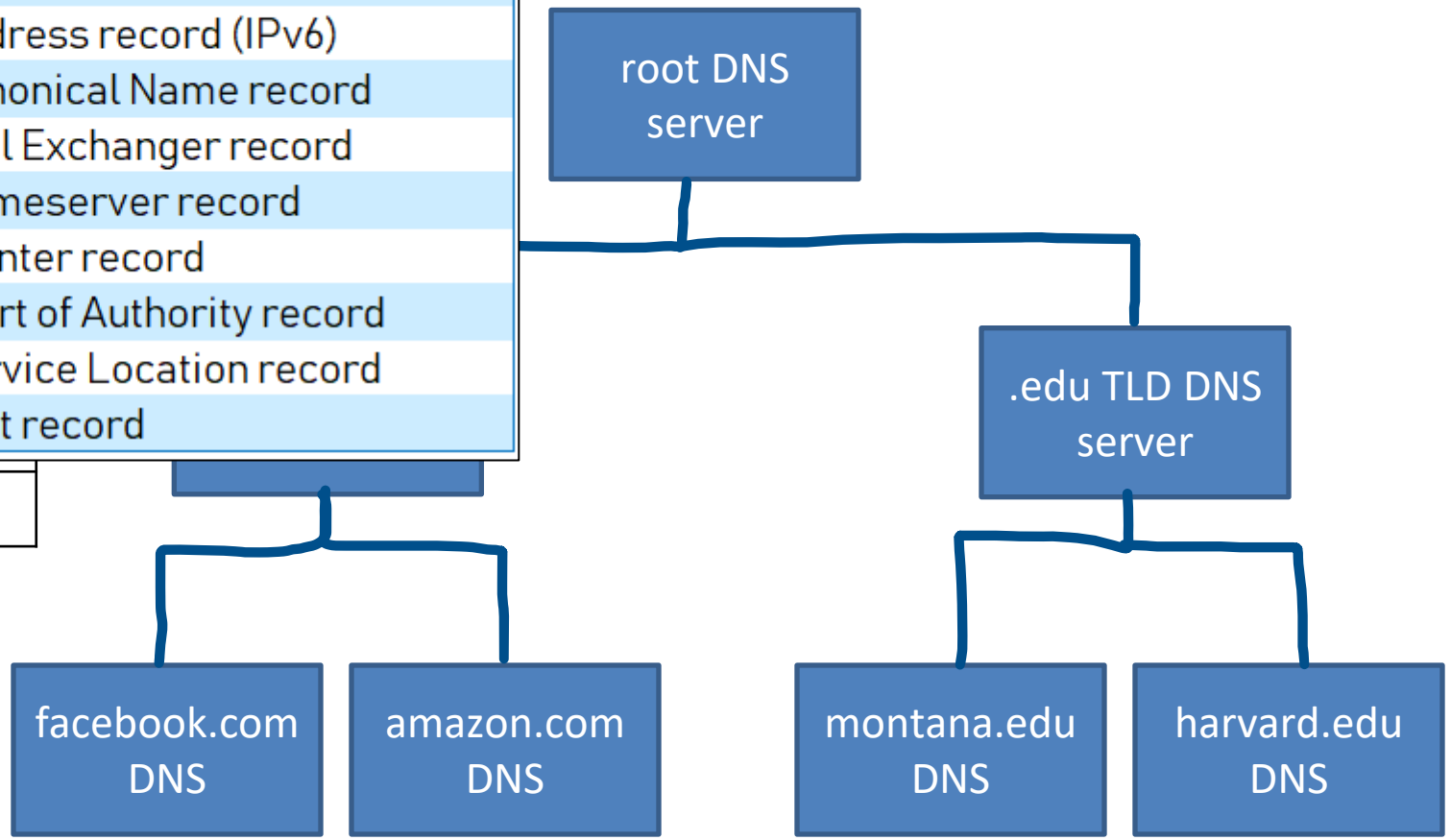
DNS Architecture

- DNS is a **distrib** server has all the records!)

Hostname	IP Ad
.com TLD DNS server	21.220
.org TLD DNS server	68.198
.edu TLD DNS server	103.10
.gov TLD DNS server	39.61.
…	…

Common DNS Record Types

Record	Description
A	Address record (IPv4)
AAAA	Address record (IPv6)
CNAME	Canonical Name record
MX	Mail Exchanger record
NS	Nameserver record
PTR	Pointer record
SOA	Start of Authority record
SRV	Service Location record
TXT	Text record



The root DNS server holds records for TLD DNS servers for all top-level domains

DNS Traffic in Wireshark

```
C:\Users\Reese Pearsall>nslookup umt.edu
Server:  dns2.msu.montana.edu
Address:  153.90.2.1

Non-authoritative answer:
Name:    umt.edu
Address:  150.131.194.46


C:\Users\Reese Pearsall>nslookup umt.edu 8.8.8.8
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    umt.edu
Address:  150.131.194.46
```

Local DNS Server

Now forcing to use 8.8.8.8

IP Whois

```
NetRange:       104.16.0.0 - 104.31.255.255
CIDR:           104.16.0.0/12
NetName:        CLOUDFLARENET
NetHandle:      NET-104-16-0-0-1
Parent:         NET104 (NET-104-0-0-0-0)
NetType:        Direct Allocation
OriginAS:       AS13335
Organization:   Cloudflare, Inc. (CLOUD14)
RegDate:        2014-03-28
Updated:        2024-09-04
Comment:        All Cloudflare abuse reporting can be done via https://www.cloudflare.com/abuse
Comment:        Geofeed: https://api.cloudflare.com/local-ip-ranges.csv
Ref:            https://rdap.arin.net/registry/ip/104.16.0.0
```

Cloudflare is a company that provides a variety of network and security services for companies

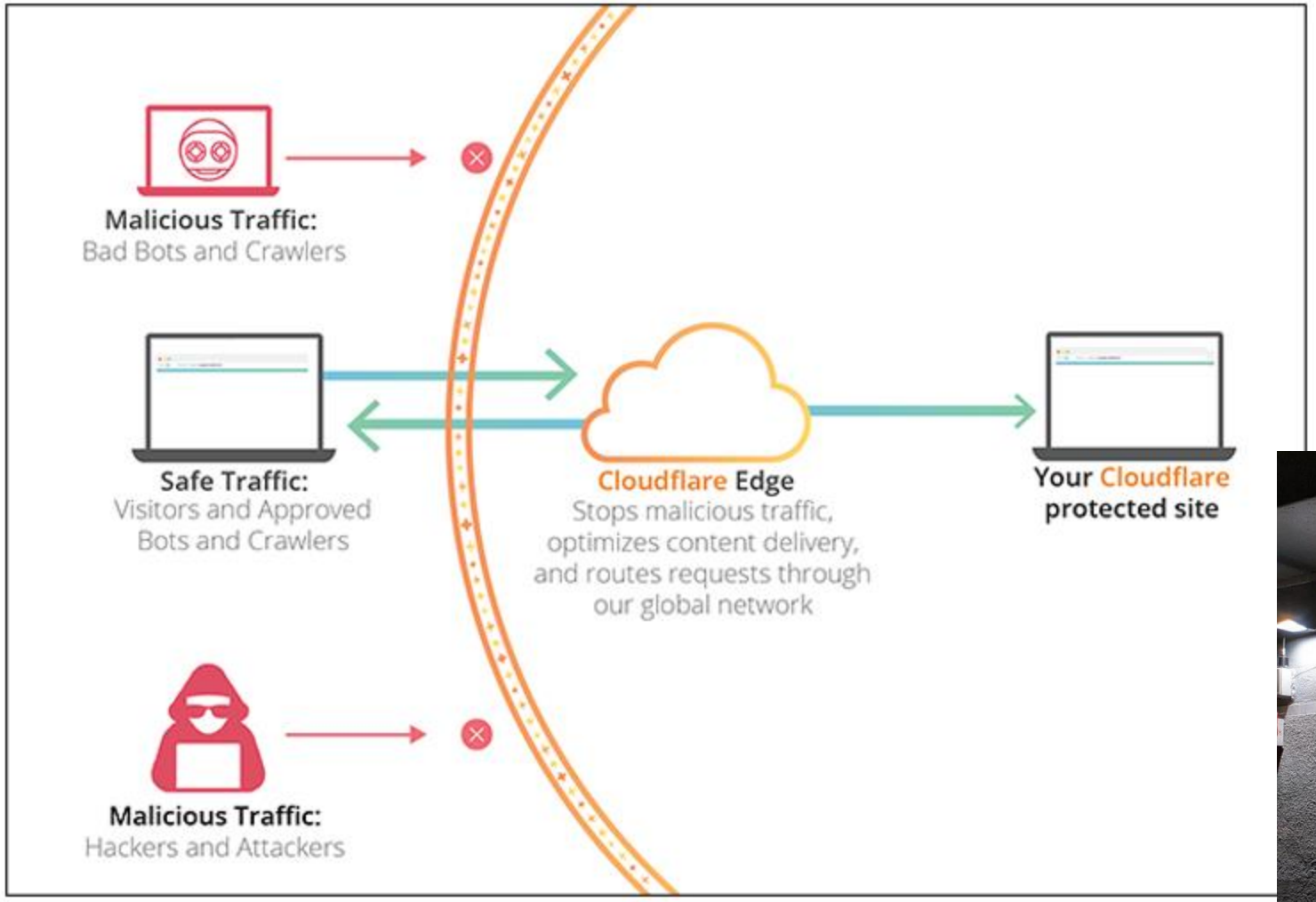
1. Content Delivery Networks (**CDN**)
- Large network of edge servers that store dynamic and static content

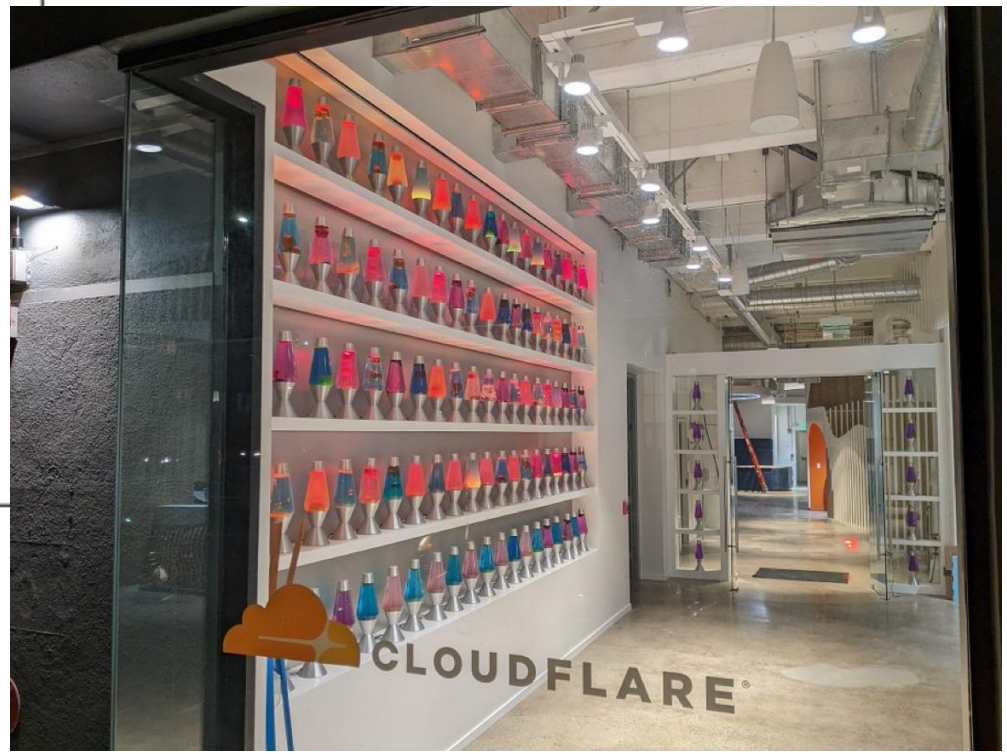
2. DNS Management
- Cloudflare will handle DNS resolutions for hostnames, websites, and nameservers

3. Web Security Management
- DDoS Protection, Firewalls, Encryption + Certificates

4. Cloud Computing
- “Cloudflare Workers” can be deployed on Cloudflare’s edge network

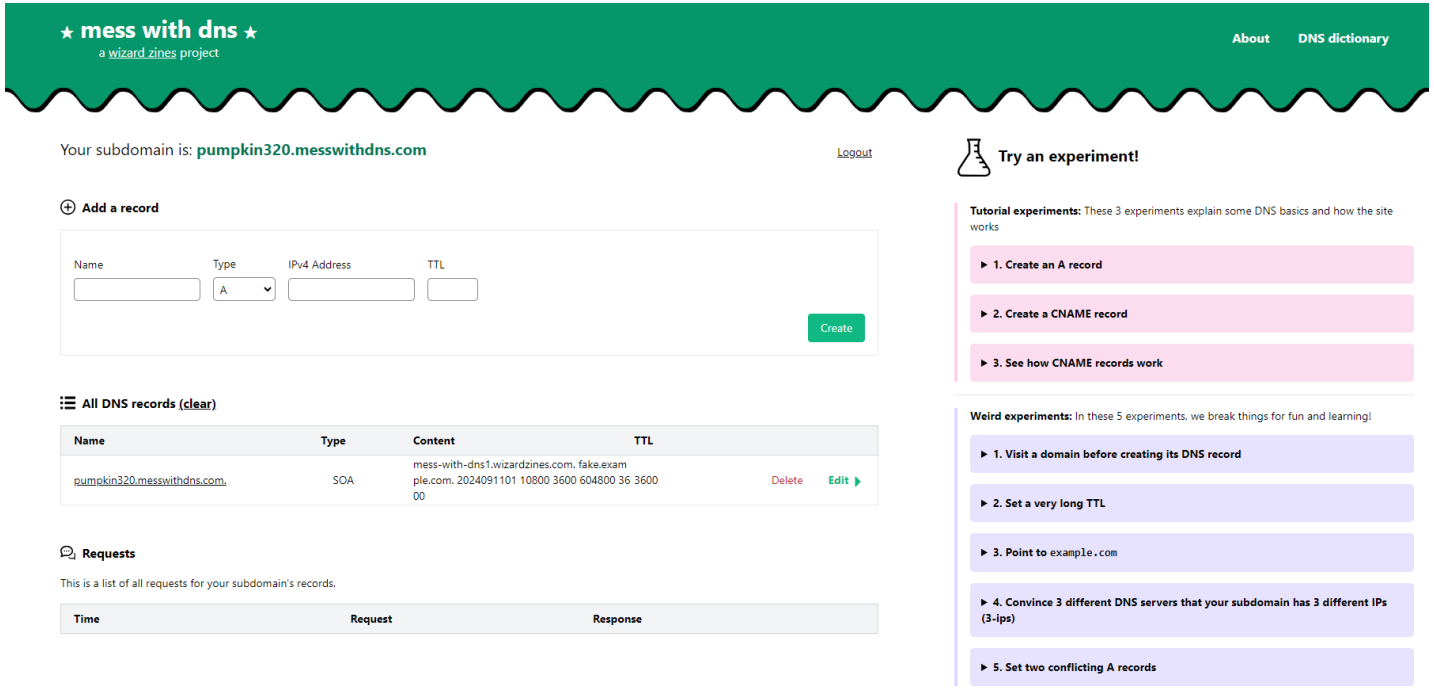
5. And much much more…


Malicious Traffic:
Bad Bots and Crawlers
Safe Traffic:
Visitors and Approved
Bots and Crawlers
Cloudflare Edge
Stops malicious traffic,
optimizes content delivery,
and routes requests through
our global network
Your Cloudflare
protected site
Malicious Traffic:
Hackers and Attackers


CLOUDFLARE

Mess with DNS

An interesting DNS “sandbox” that allows you to create a mock DNS server with your own records

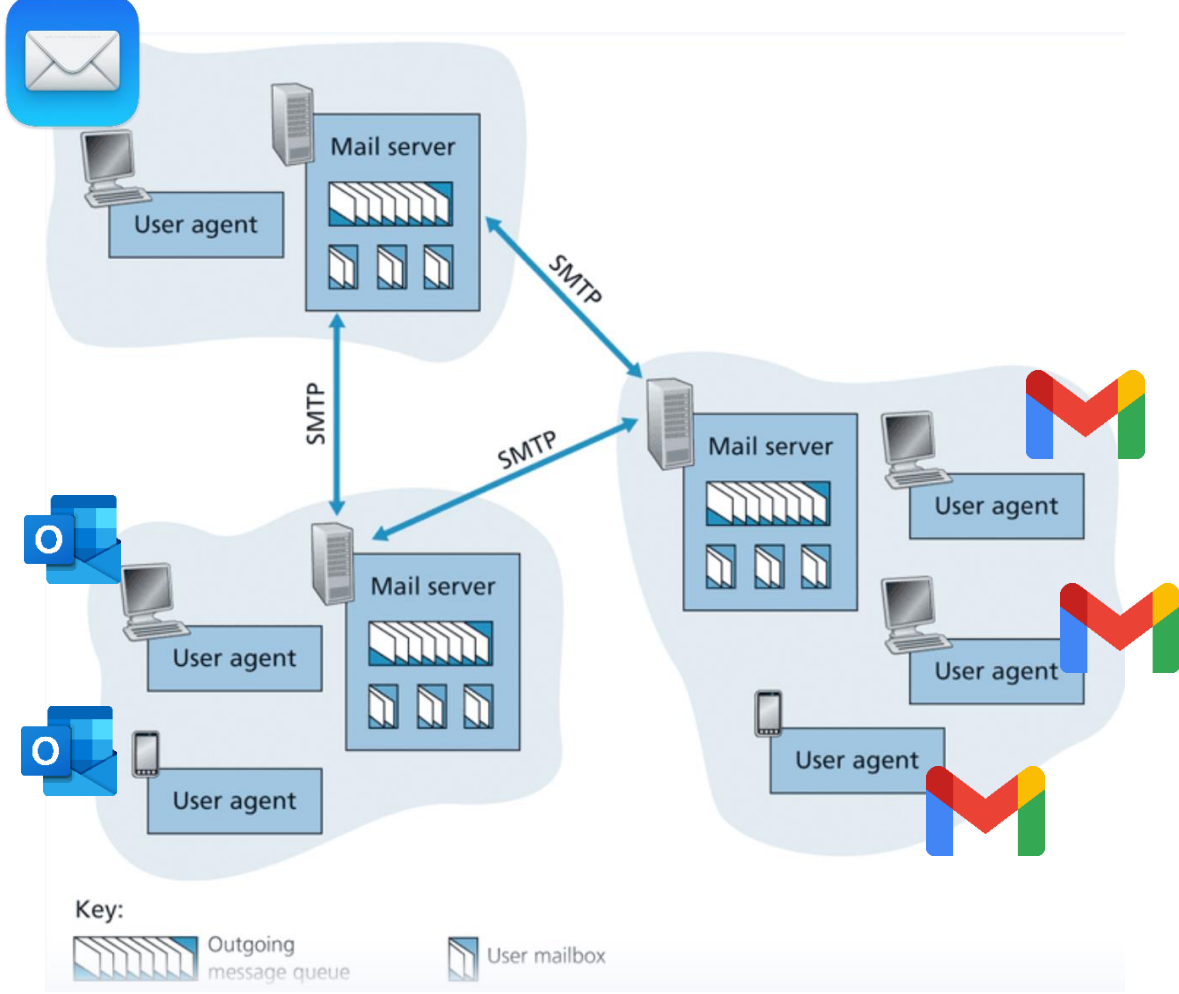


SMTP

Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another

SMTP

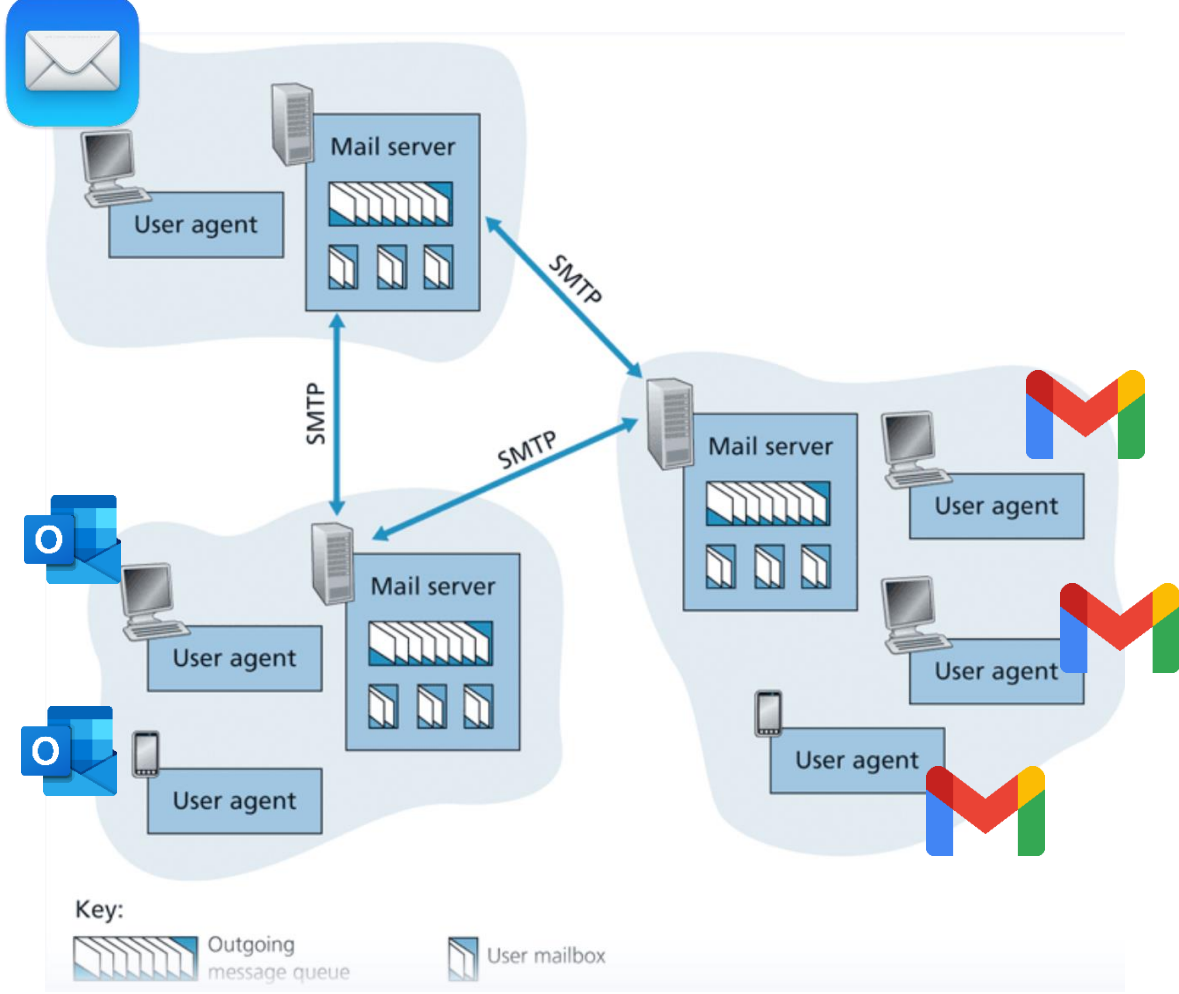
Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another



SMTP

Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another

Each recipient has a **mailbox** location in one of the mail servers



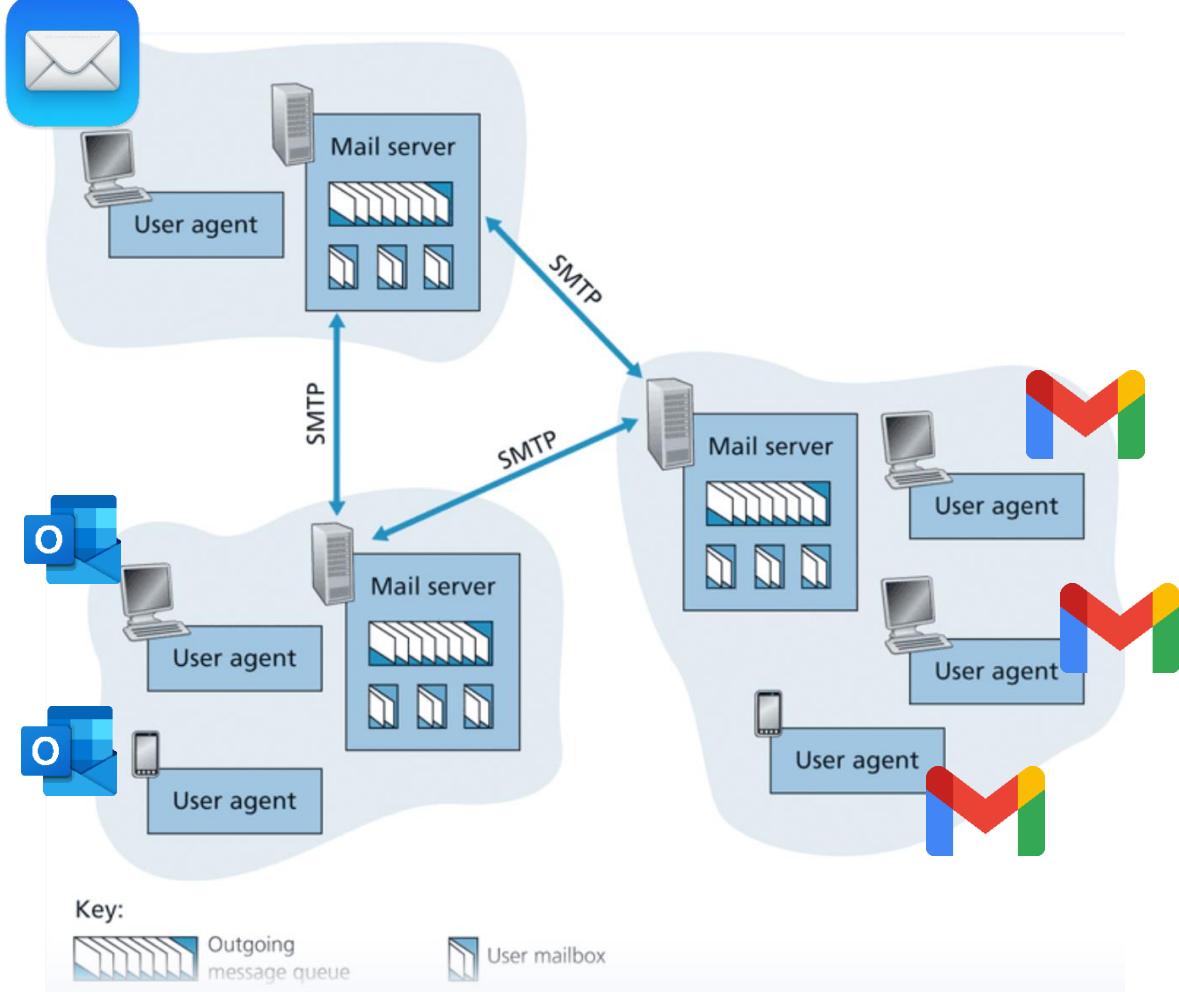
SMTP

Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another

Each recipient has a **mailbox** location in one of the mail servers

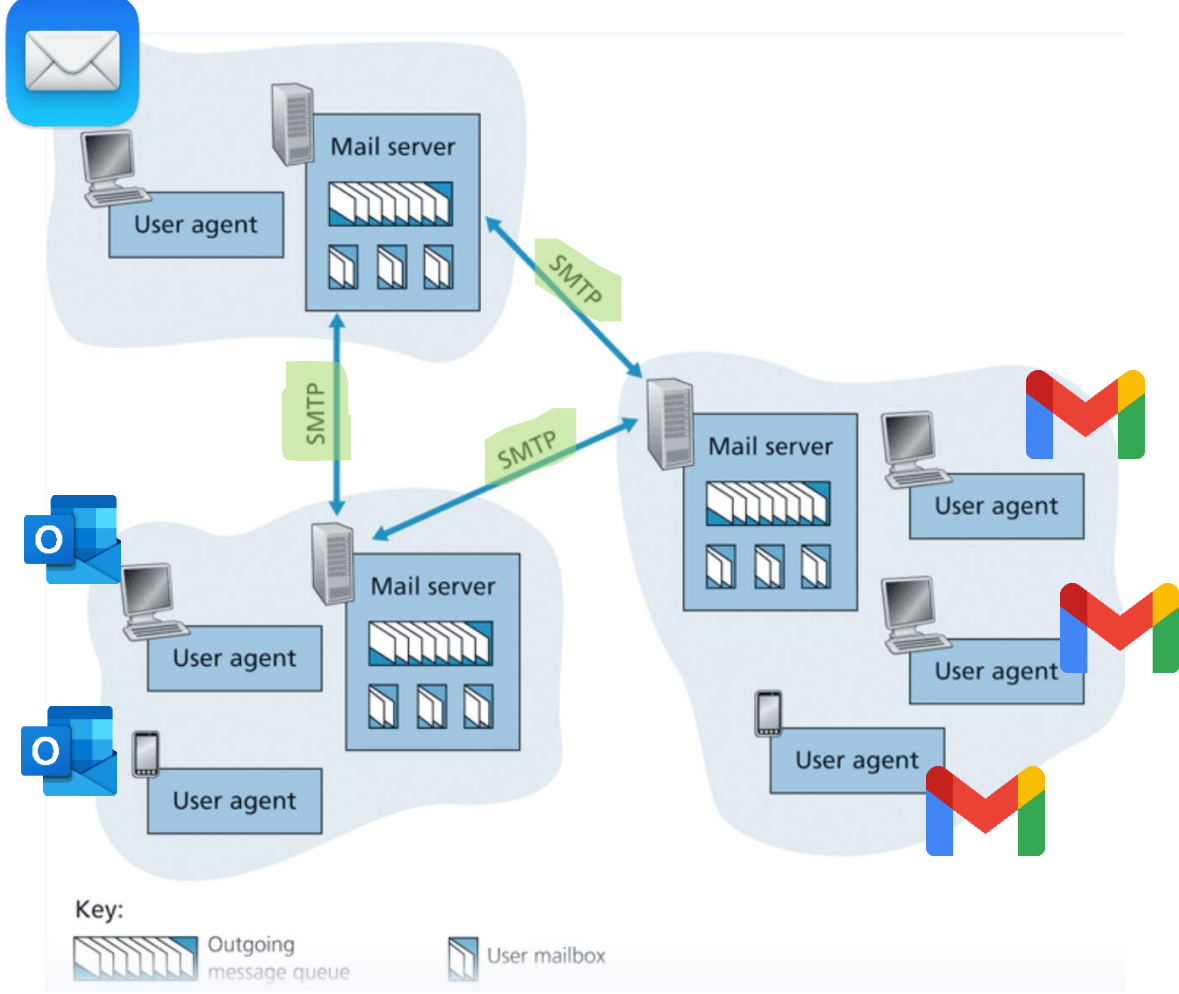
Messages are put in an outgoing **message queue** when they are sent

SMTP uses **TCP** to ensure reliable data transfer of emails

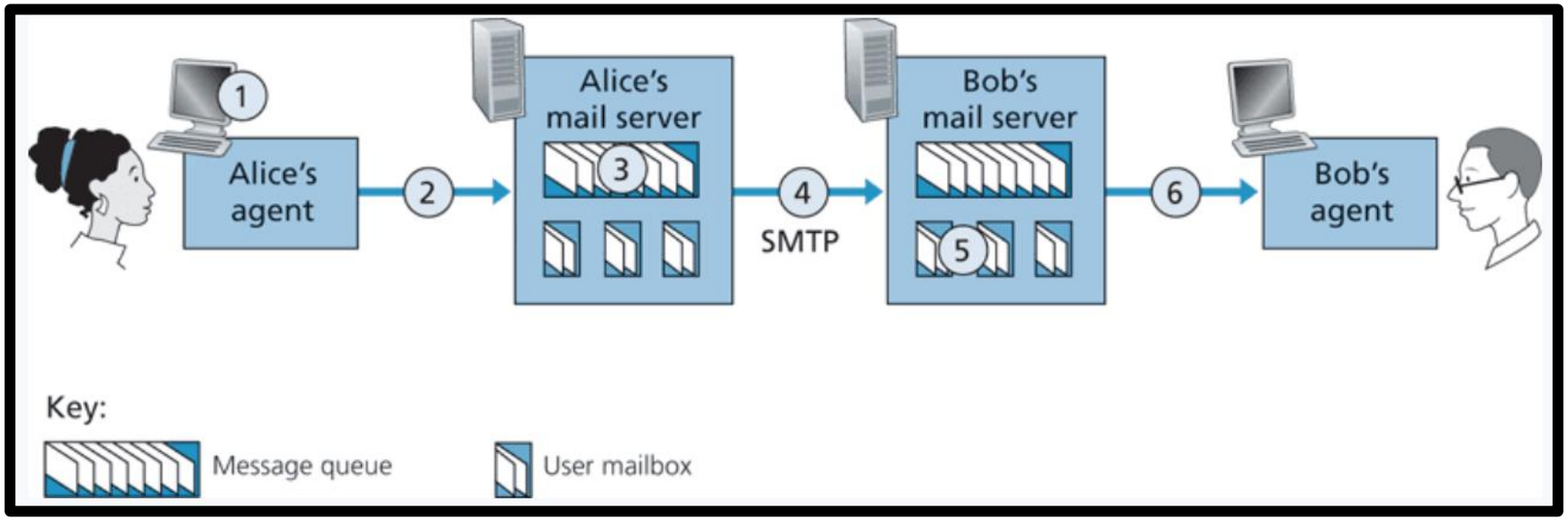


SMTP

Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another

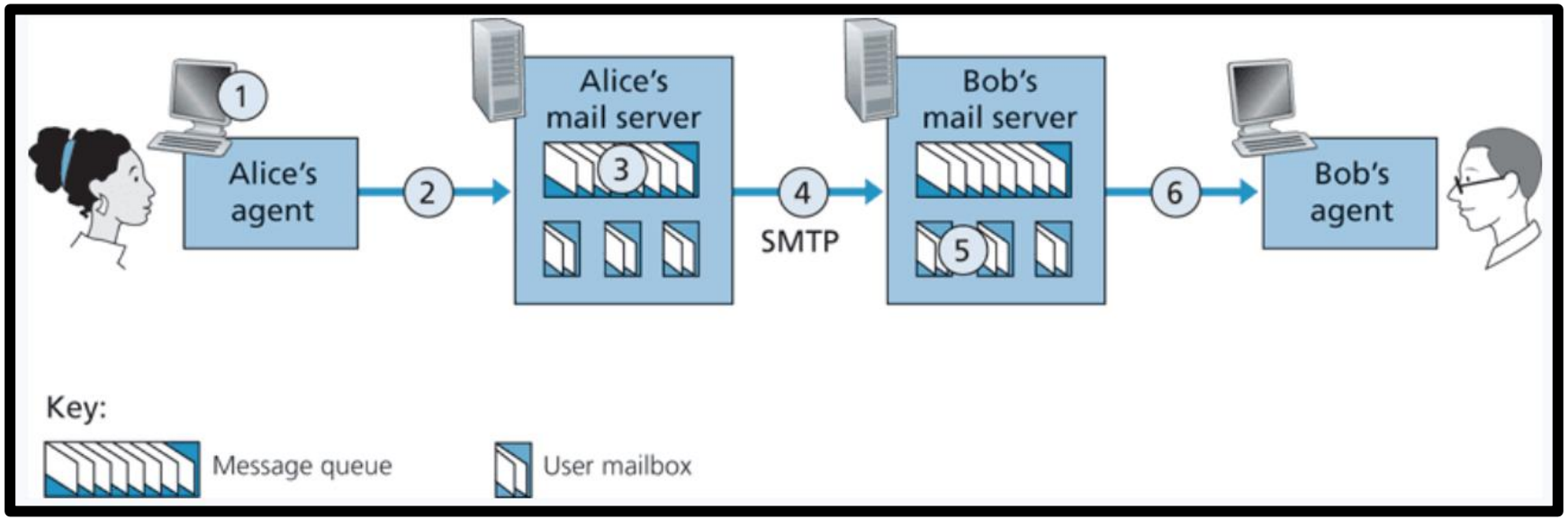


SMTP



1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.

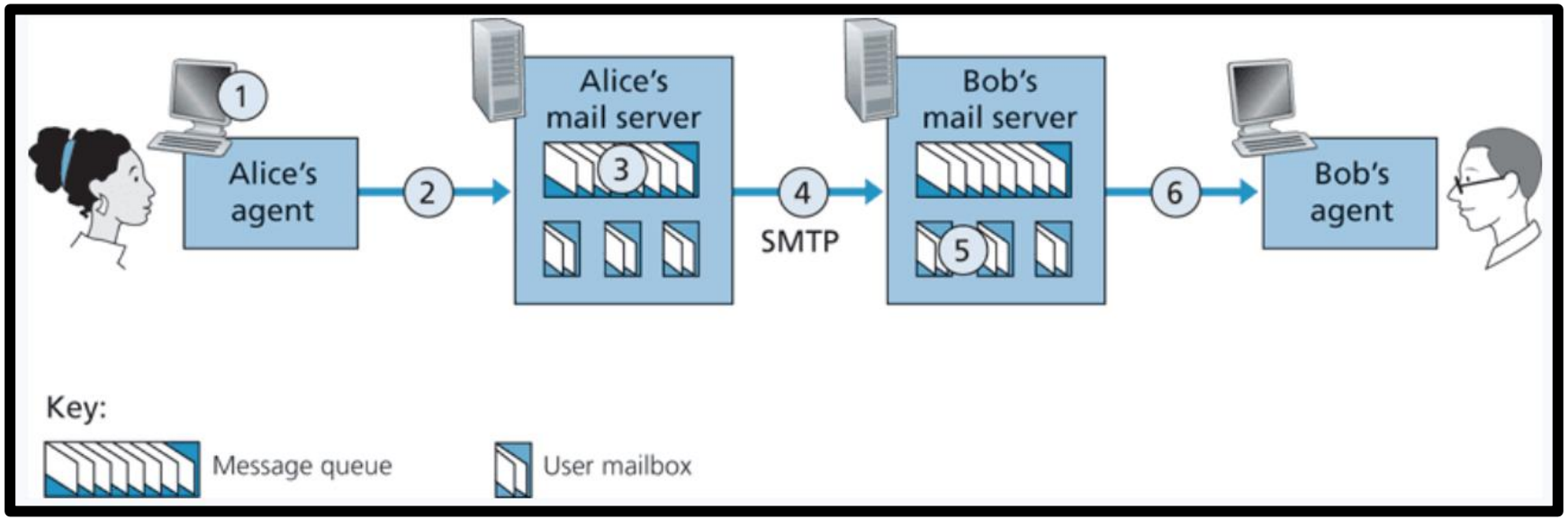
SMTP



1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.

2. Alice's user agent sends the message to her mail server, where it is placed in a message queue.

SMTP

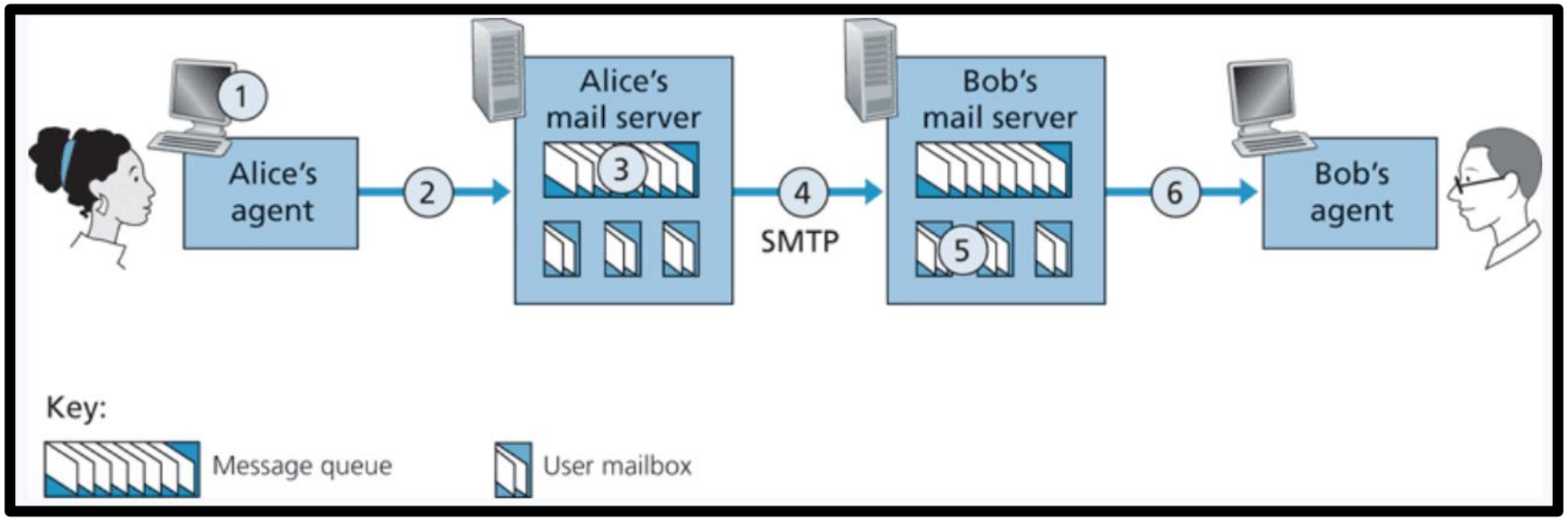


1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.

2. Alice's user agent sends the message to her mail server, where it is placed in a message queue.

3. The client side of SMTP, running on Alice's mail server, sees the message in the message queue. It opens a TCP connection to an SMTP server, running on Bob's mail server.

SMTP



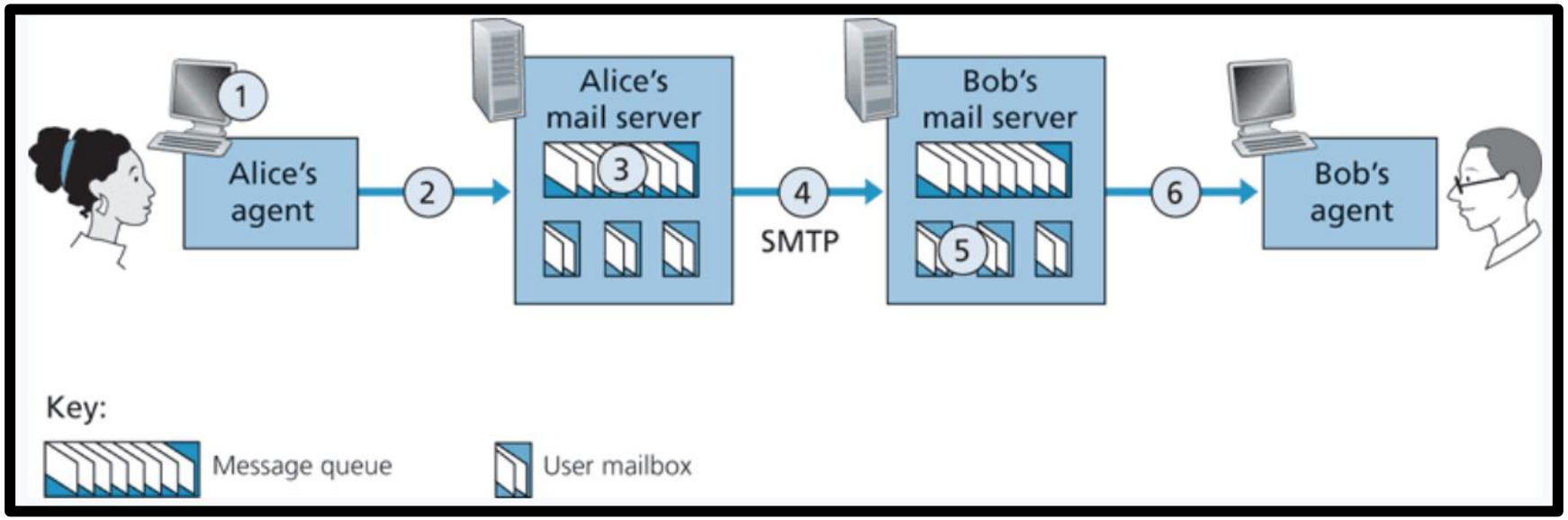
1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.

2. Alice's user agent sends the message to her mail server, where it is placed in a message queue.

3. The client side of SMTP, running on Alice's mail server, sees the message in the message queue. It opens a TCP connection to an SMTP server, running on Bob's mail server.

4. After some initial SMTP handshaking, the SMTP client sends Alice's message into the TCP connection.

SMTP



1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.

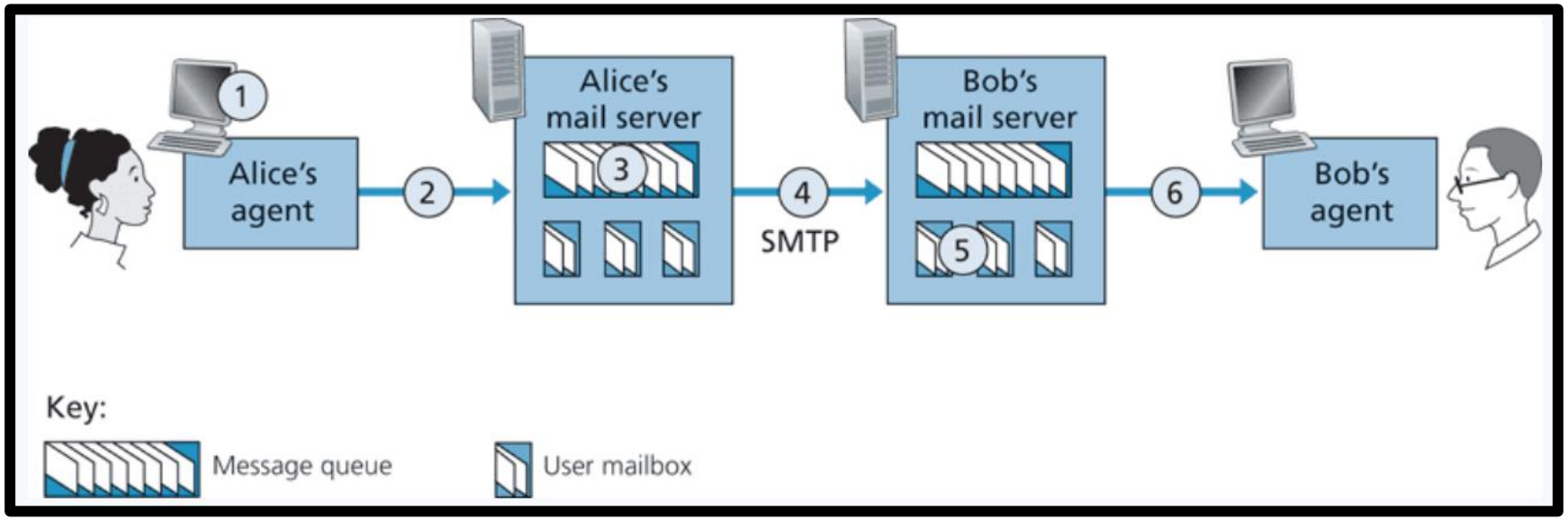
2. Alice's user agent sends the message to her mail server, where it is placed in a message queue.

3. The client side of SMTP, running on Alice's mail server, sees the message in the message queue. It opens a TCP connection to an SMTP server, running on Bob's mail server.

4. After some initial SMTP handshaking, the SMTP client sends Alice's message into the TCP connection.

5. At Bob's mail server, the server side of SMTP receives the message. Bob's mail server then places the message in Bob's mailbox.

SMTP



1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.

2. Alice's user agent sends the message to her mail server, where it is placed in a message queue.

3. The client side of SMTP, running on Alice's mail server, sees the message in the message queue. It opens a TCP connection to an SMTP server, running on Bob's mail server.

4. After some initial SMTP handshaking, the SMTP client sends Alice's message into the TCP connection.

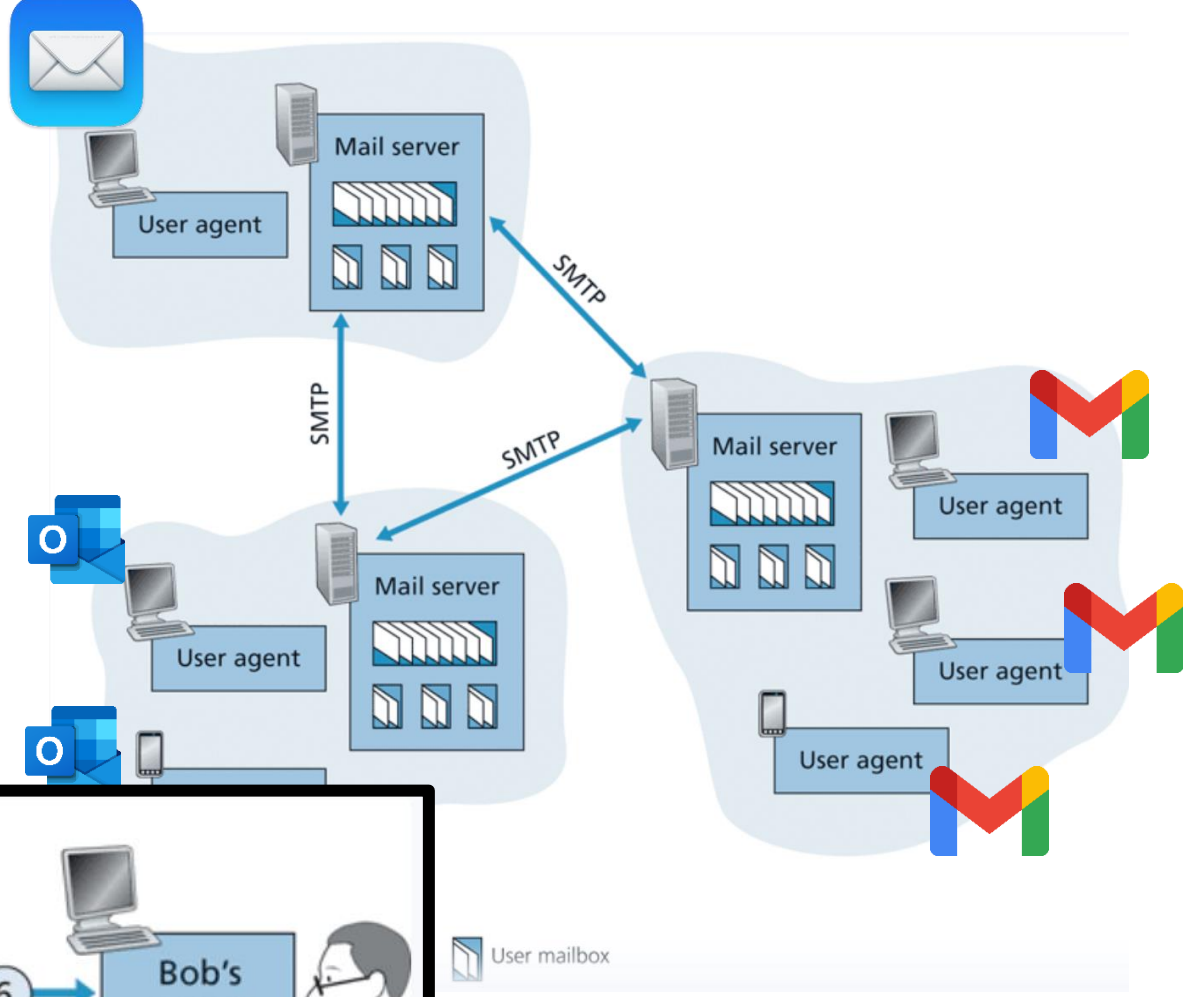
5. At Bob's mail server, the server side of SMTP receives the message. Bob's mail server then places the message in Bob's mailbox.

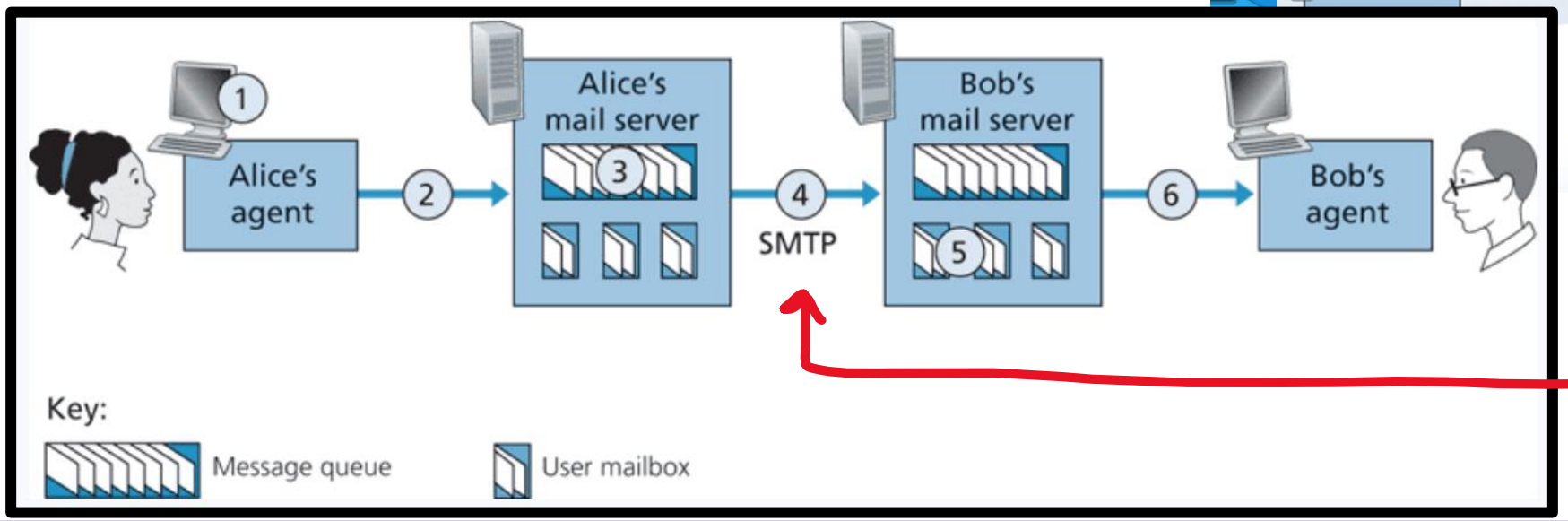
6. Bob invokes his user agent to read the message at his convenience.

SMTP

Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another

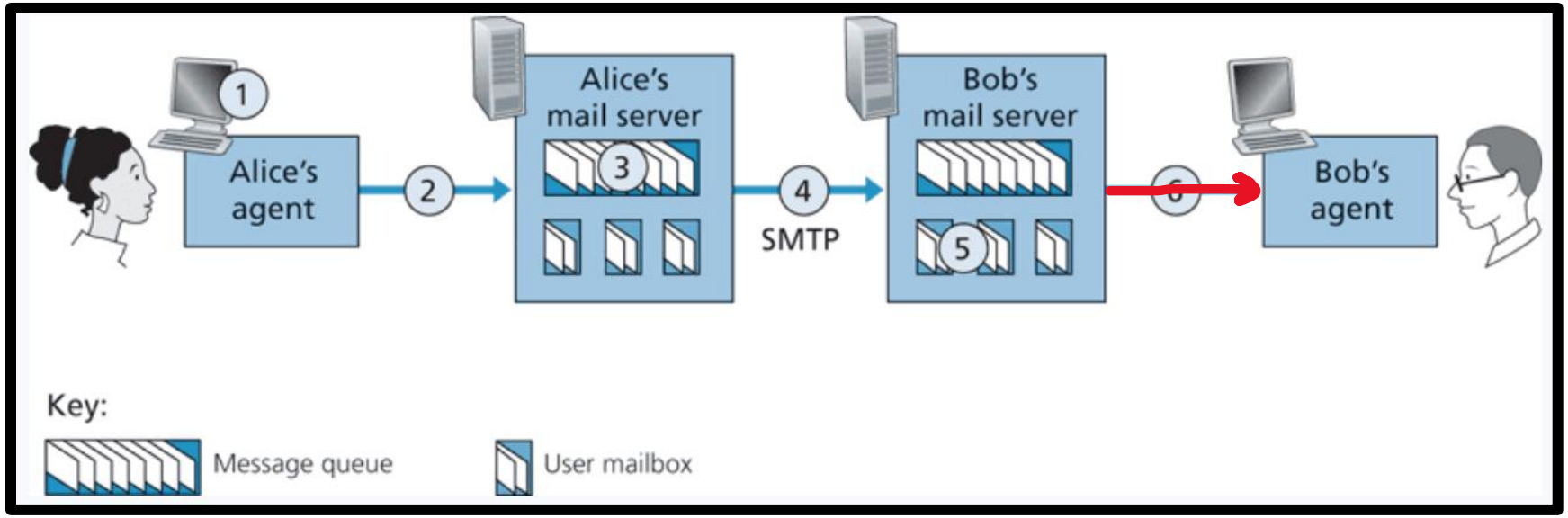
This is not a protocol for *retrieving* emails





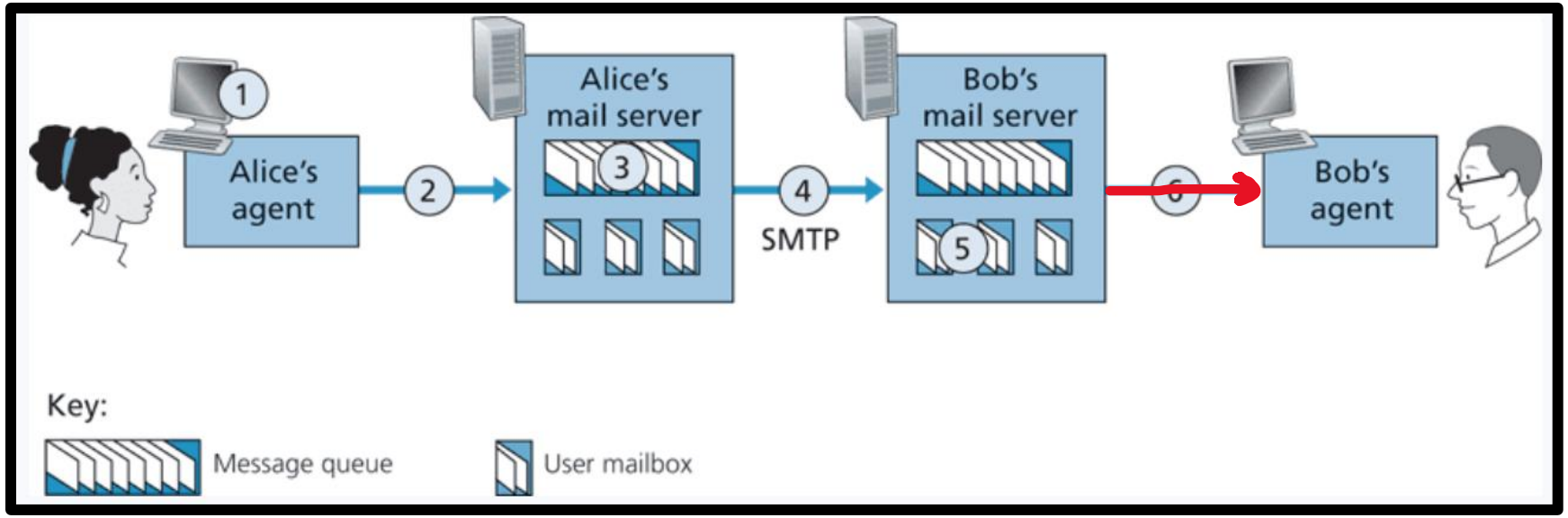
STMP uses TCP for the end-to-end delivery (**DIRECT**) **(PORT 25)**

SMTP



POP3 (post office protocol) or **IMAP** (internet message access protocol) are used to retrieve emails from mail servers.

SMTP



POP3 (post office protocol) or **IMAP** (internet message access protocol) are used to retrieve emails from mail servers.

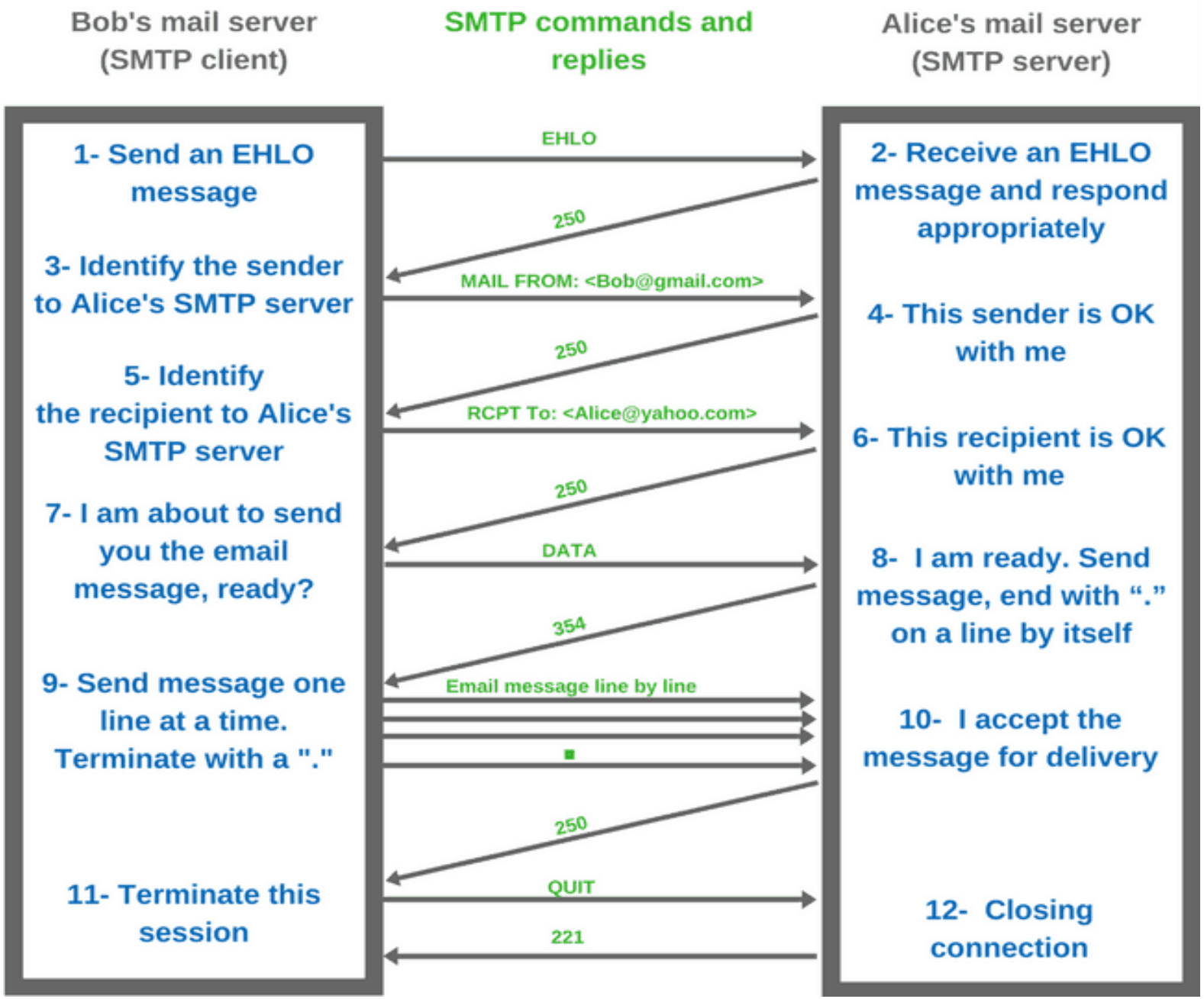
POP3 deletes the email of the web server, IMAP maintains a copy to synchronize across multiple devices

SMTP

SMTP Handshake + Message exchange format

(Very verbose)

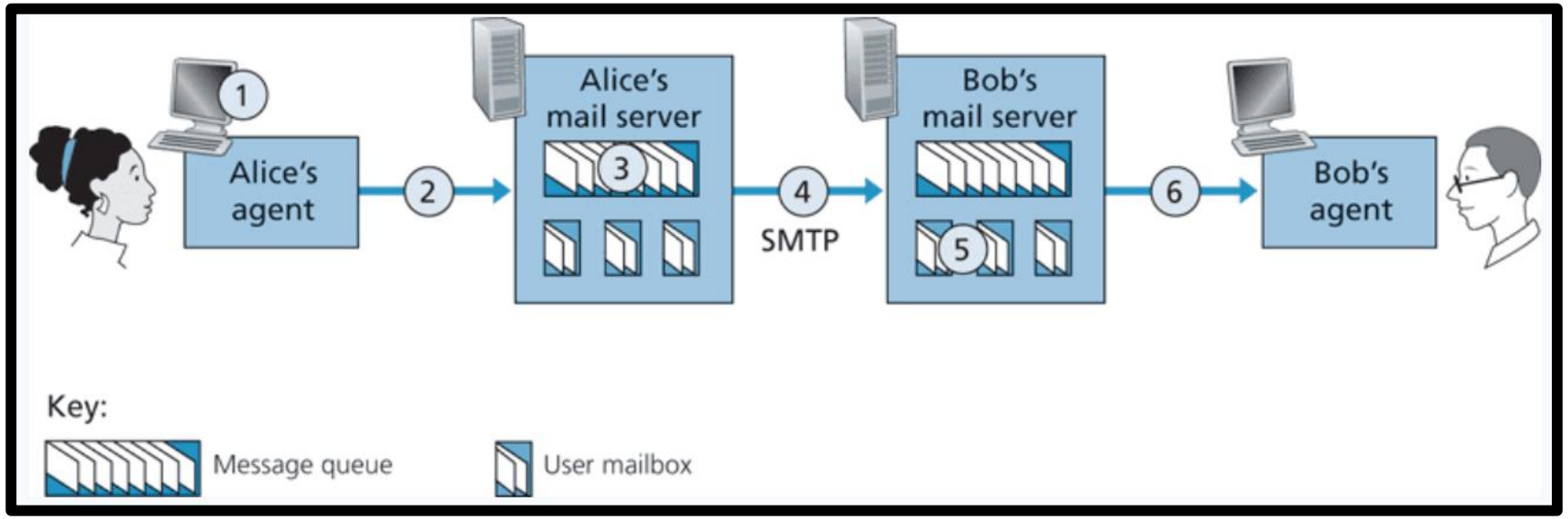
```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with ”.” on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```


Bob's mail server
(SMTP client)
SMTP commands and replies
Alice's mail server
(SMTP server)
1- Send an EHLO message
EHLO
2- Receive an EHLO message and respond appropriately
250
3- Identify the sender to Alice's SMTP server
MAIL FROM: <Bob@gmail.com>
4- This sender is OK with me
250
5- Identify the recipient to Alice's SMTP server
RCPT To: <Alice@yahoo.com>
6- This recipient is OK with me
250
7- I am about to send you the email message, ready?
DATA
8- I am ready. Send message, end with "." on a line by itself
354
9- Send message one line at a time. Terminate with a "."
Email message line by line
.
10- I accept the message for delivery
250
11- Terminate this session
QUIT
12- Closing connection
221

SMTP

Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another *asynchronously*

Ports 25 is reserved for SMTP traffic (and also port 587 & 465)



SMTP Traffic in Wireshark