CSCI 466: Networks

Lecture 8: Application Layer

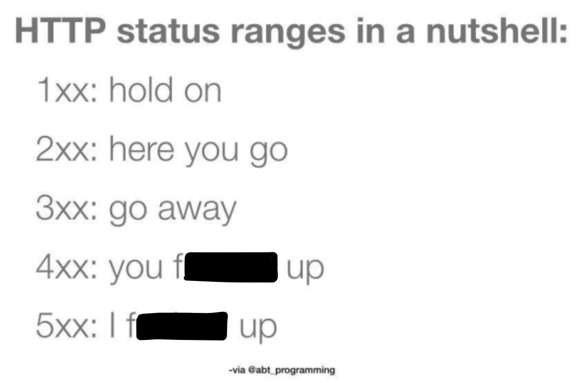
(More DNS, SMTP, FTP)

Reese Pearsall

Fall 2023

Announcements

Wireshark Lab 1 due on Wednesday @ 11:59 PM



HTTP status ranges in a nutshell:

1xx: hold on

2xx: here you go

3xx: go away

4xx: you f[redacted] up

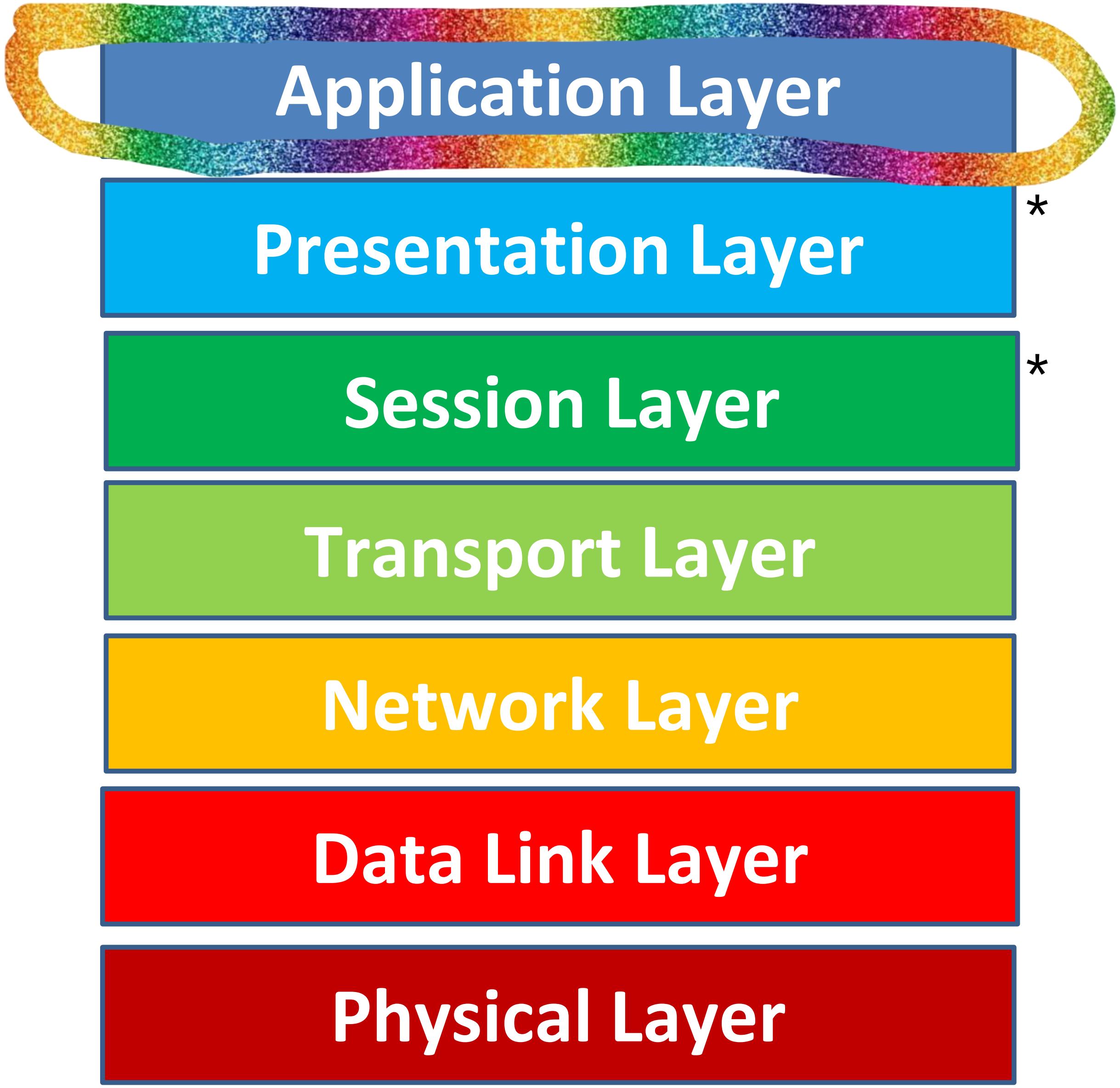
5xx: I f[redacted] up

-via @abt_programming

ACM and AWC will be hosting a career fair prep workshop tomorrow at 5:00 PM

- Practice Interviews
- Career fair opportunities and companies
- How to prepare
- Casual resume review

OSI Model



Application Layer

Presentation Layer *

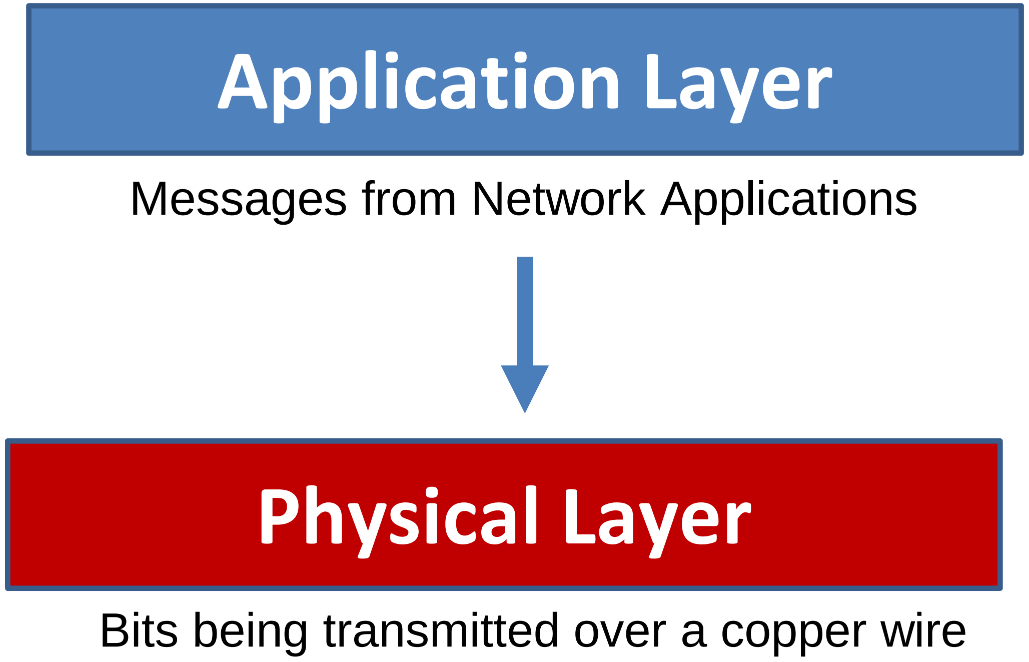
Session Layer *

Transport Layer

Network Layer

Data Link Layer

Physical Layer



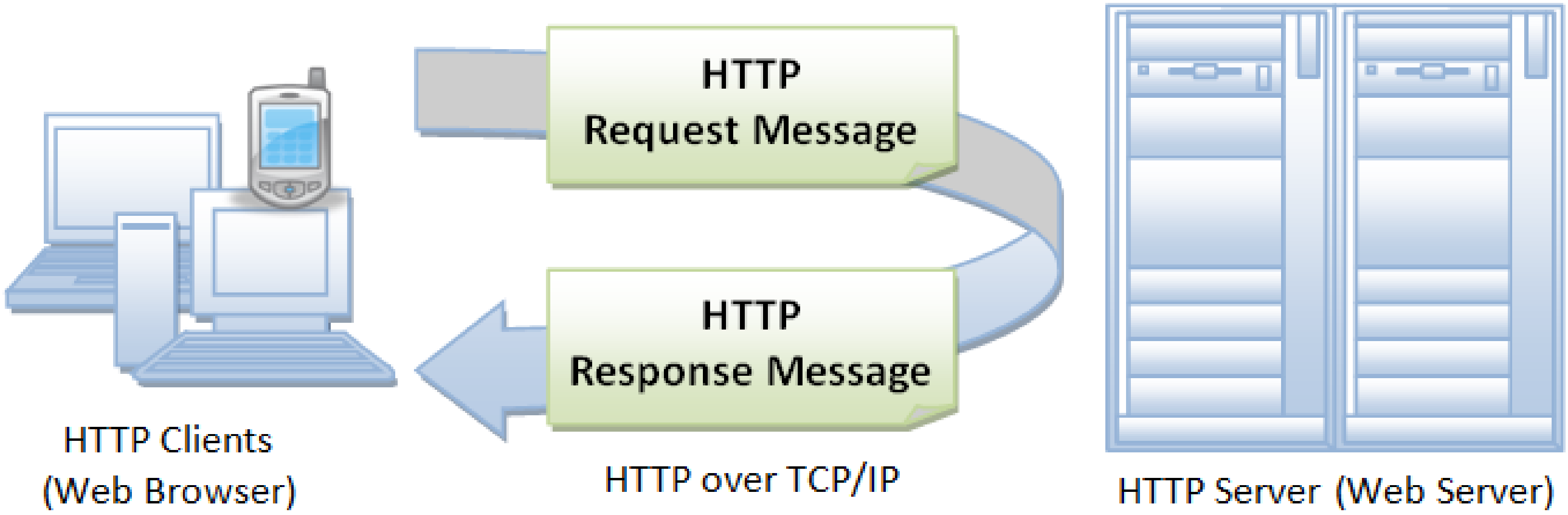
Application Layer

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

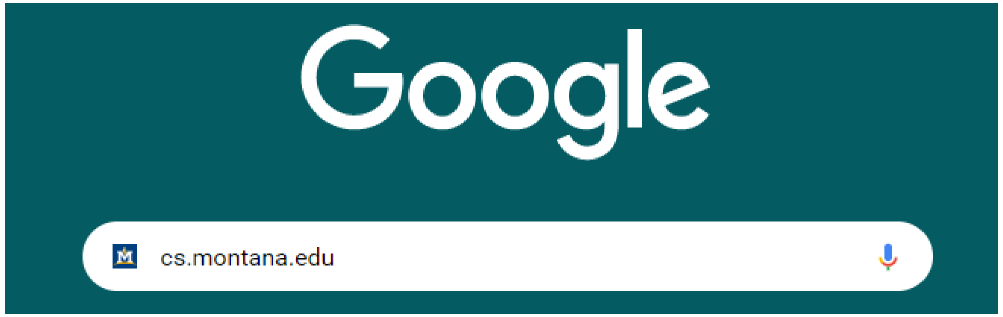


HTTP Requests in Python

DNS

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

Computers understand numbers
- (They need IP addresses)



cs.montana.edu

→ **DNS** → 153.90.127.197

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

DNS Architecture

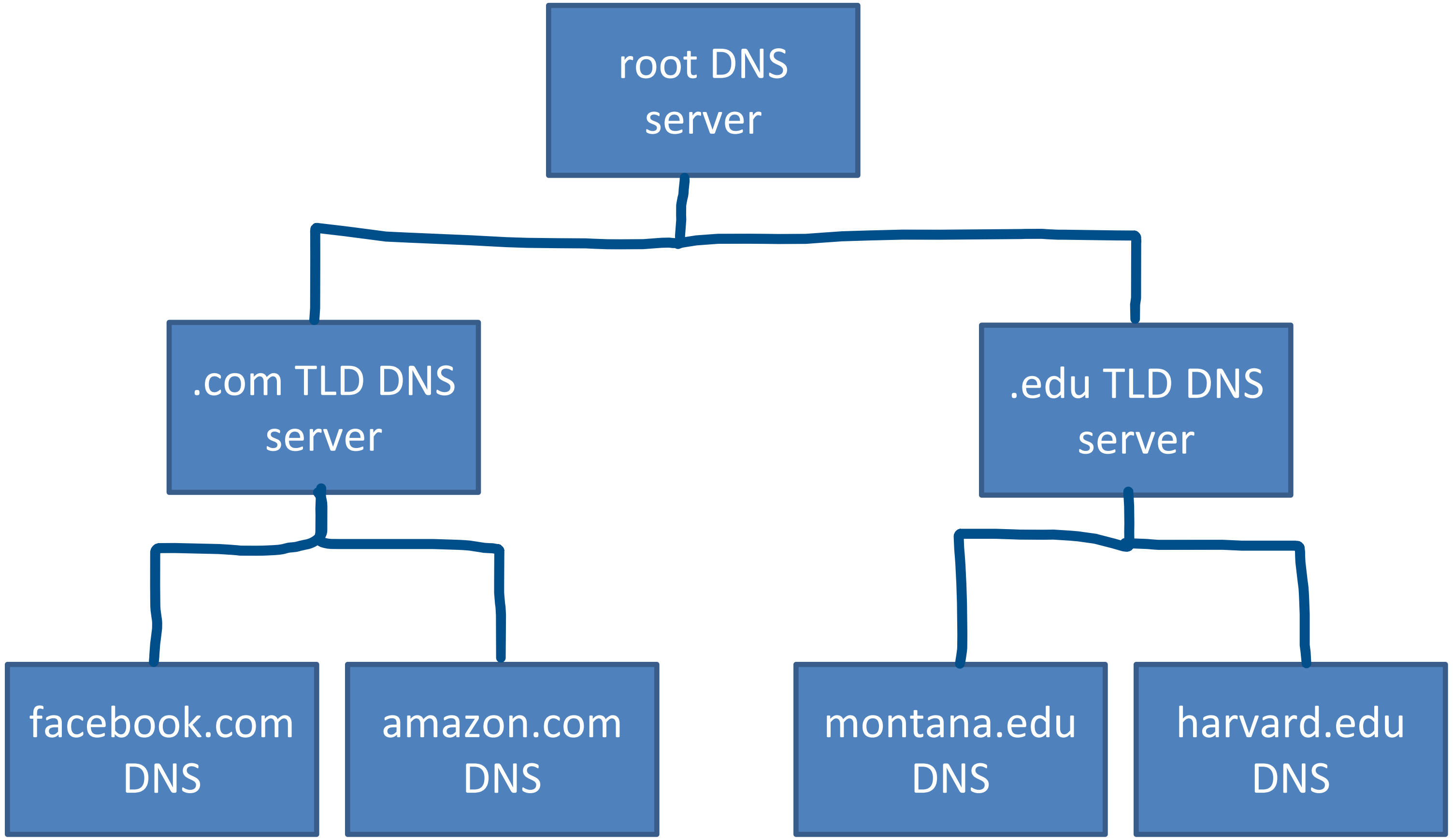
- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

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)

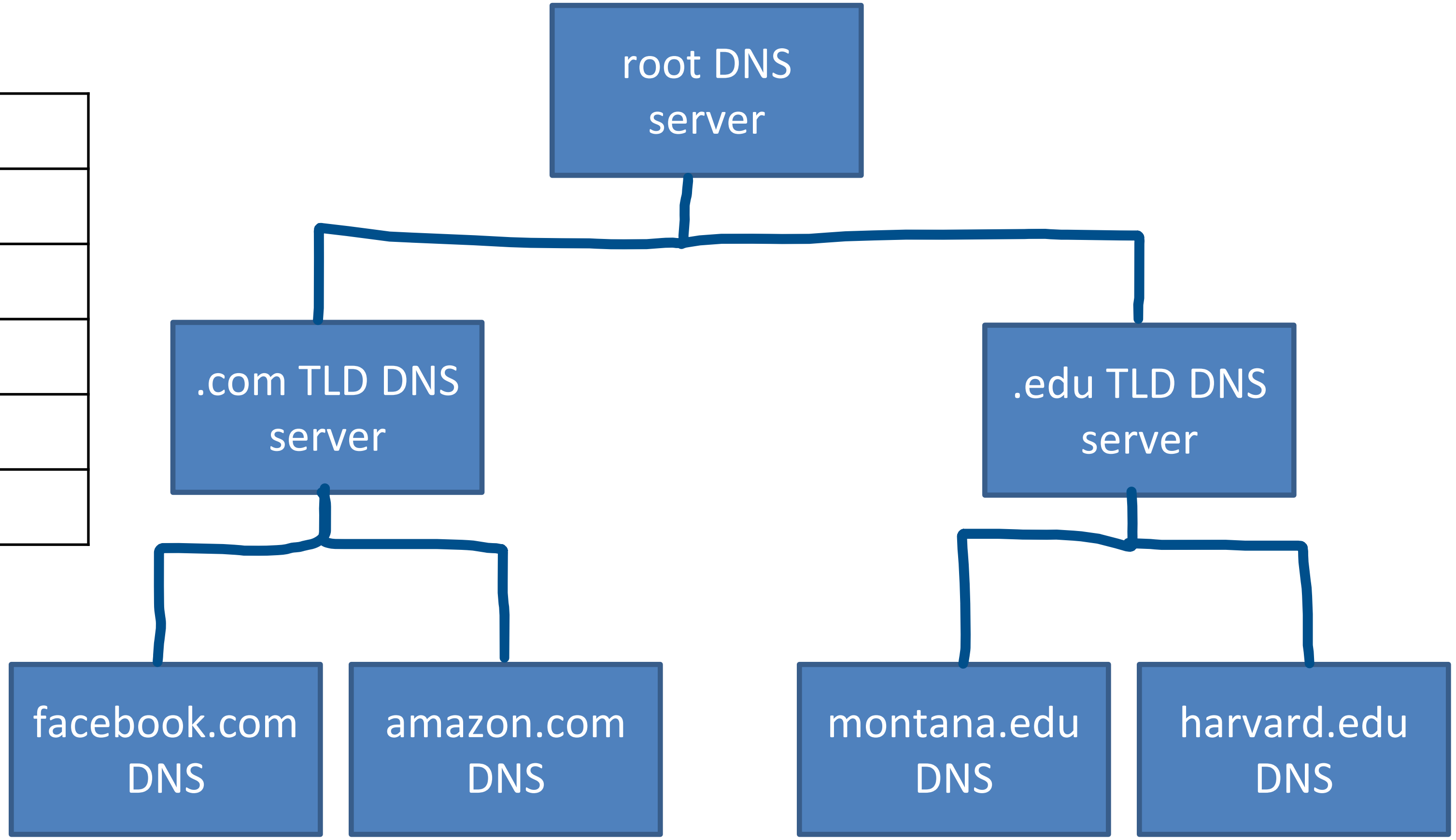
Root DNS servers- responsible for maintaining IP addresses for TLD servers



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
…	…

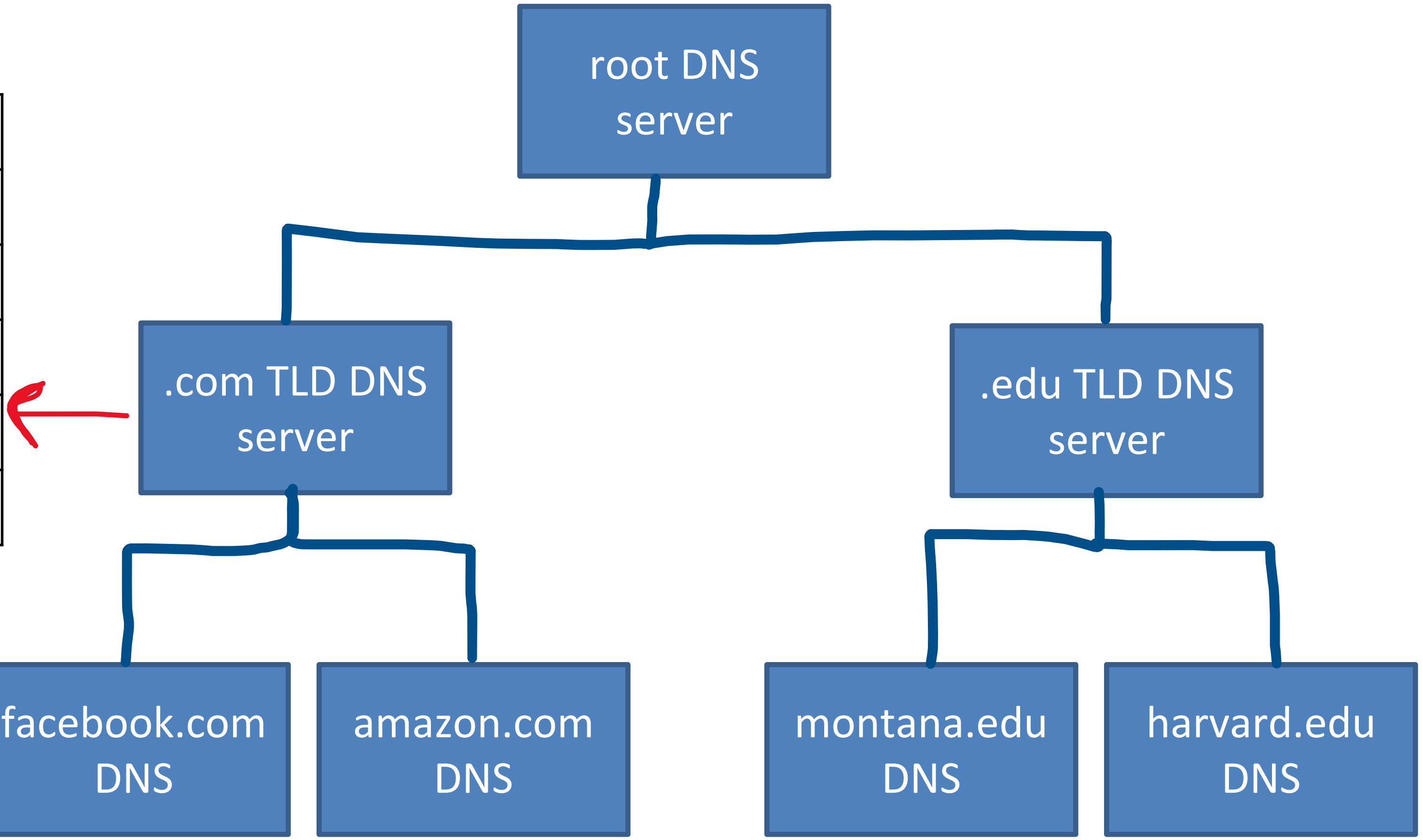


Authoritative DNS Server

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
…	…

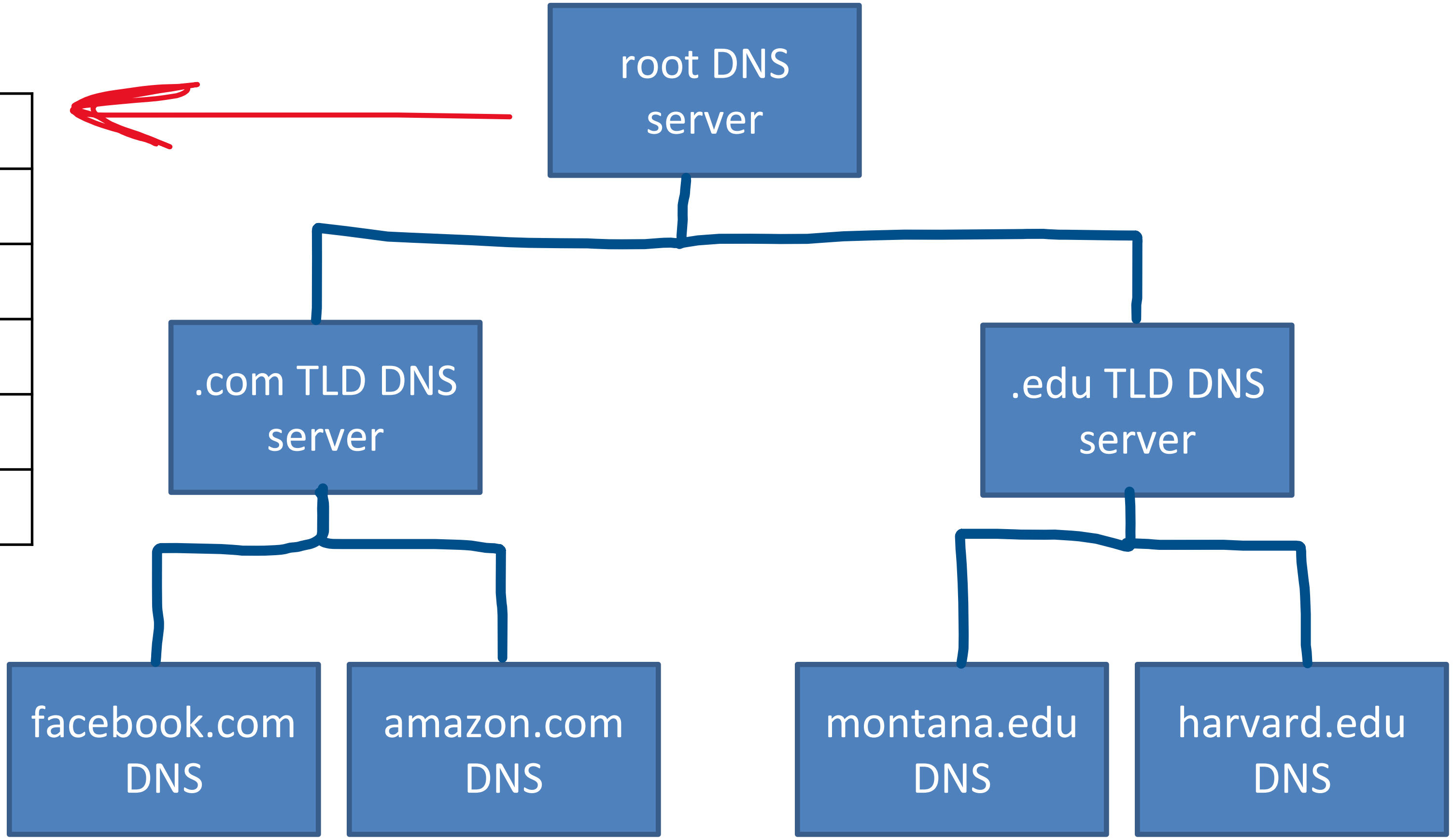


TLD DNS servers hold records for authoritative DNS server for a particular domain

DNS Architecture

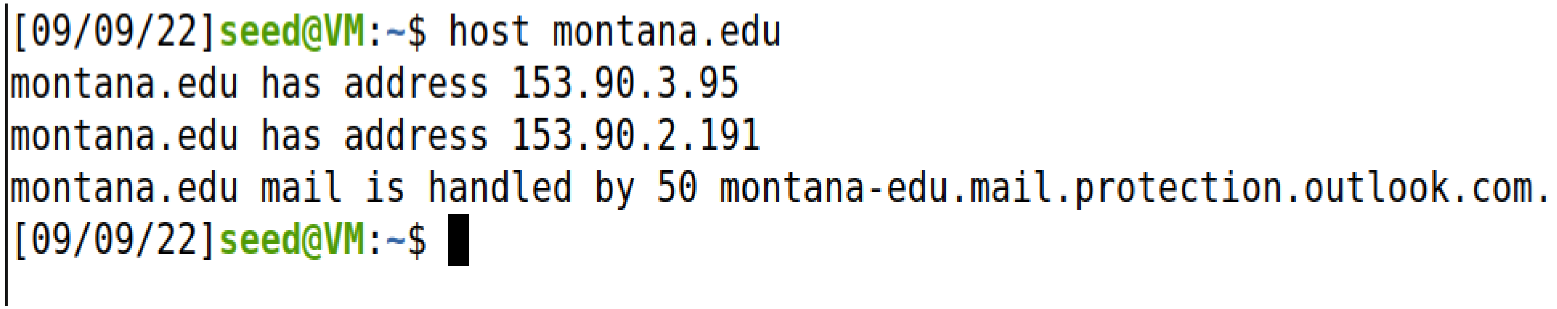
- DNS is a **distributed**, **hierarchical** database (no DNS server has all the records!)

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 Commands



```
[09/09/22]seed@VM:~$ host montana.edu
montana.edu has address 153.90.3.95
montana.edu has address 153.90.2.191
montana.edu mail is handled by 50 montana-edu.mail.protection.outlook.com.
[09/09/22]seed@VM:~$ 
```

- DNS services
  - Hostname to IP address translation

    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

```
  host huffpost.com | grep "address" | sed -n -e
's/^.*address //p'
```

  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`

153.90.3.95

(`nslookup` also works)

DNS Commands

- DNS services
  - Hostname to IP address translation

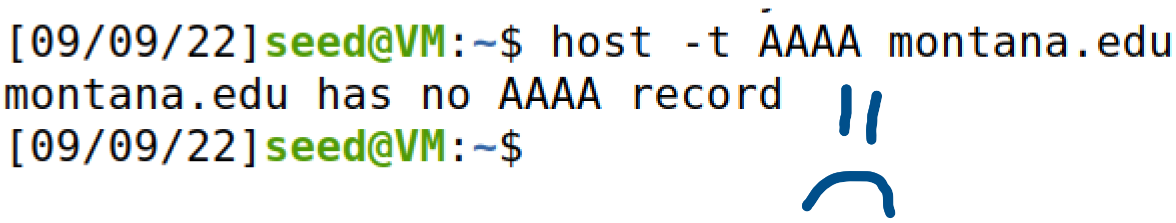
    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

    ```
    host huffpost.com | grep "address" | sed -n -e
    's/^.*address //p'
    ```
  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`



```
[09/09/22]seed@VM:~$ host -t AAAA montana.edu
montana.edu has no AAAA record
[09/09/22]seed@VM:~$
```

DNS Commands

- DNS services
  - Hostname to IP address translation

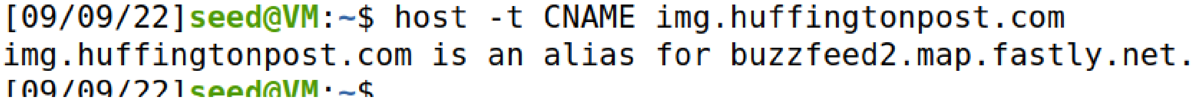
    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

    `host huffpost.com | grep "address" | sed -n -e 's/^.*address //p'`
  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`



```
[09/09/22]seed@VM:~$ host -t CNAME img.huffingtonpost.com
img.huffingtonpost.com is an alias for buzzfeed2.map.fastly.net.
[09/09/22]seed@VM:~$
```

DNS Commands

- DNS services
  - Hostname to IP address translation

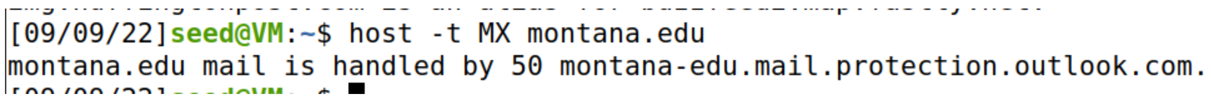
    ```
    host montana.edu
    ```

  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    ```
    host -t CNAME img.huffingtonpost.com
    ```

  - Mail server aliasing

    ```
    host -t MX montana.edu
    ```



```
[09/09/22]seed@VM:~$ host -t MX montana.edu
montana.edu mail is handled by 50 montana-edu.mail.protection.outlook.com.
```

  - Load distribution

```
    host huffpost.com | grep "address" | sed -n -e
's/^.*address //p'
```

  - Redirection
    - Look up same host from servers in different regions

    ```
    host google.com 8.8.8.8
    ```

DNS Commands

- DNS services
  - Hostname to IP address translation

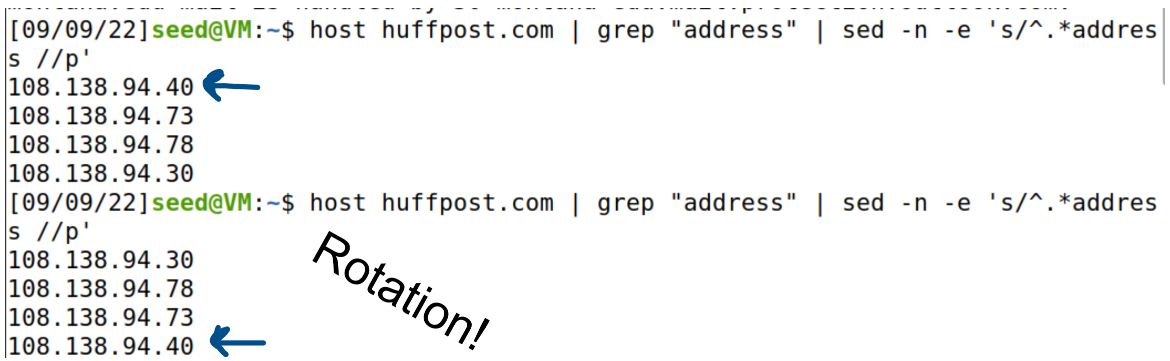
    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

    `host huffpost.com | grep "address" | sed -n -e 's/^.*address //p'`
  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`



```
[09/09/22]seed@VM:~$ host huffpost.com | grep "address" | sed -n -e 's/^.*addres
s //p'
108.138.94.40
108.138.94.73
108.138.94.78
108.138.94.30
[09/09/22]seed@VM:~$ host huffpost.com | grep "address" | sed -n -e 's/^.*addres
s //p'
108.138.94.30
108.138.94.78
108.138.94.73
108.138.94.40
```

Rotation!

DNS Commands

- DNS services
  - Hostname to IP address translation

    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

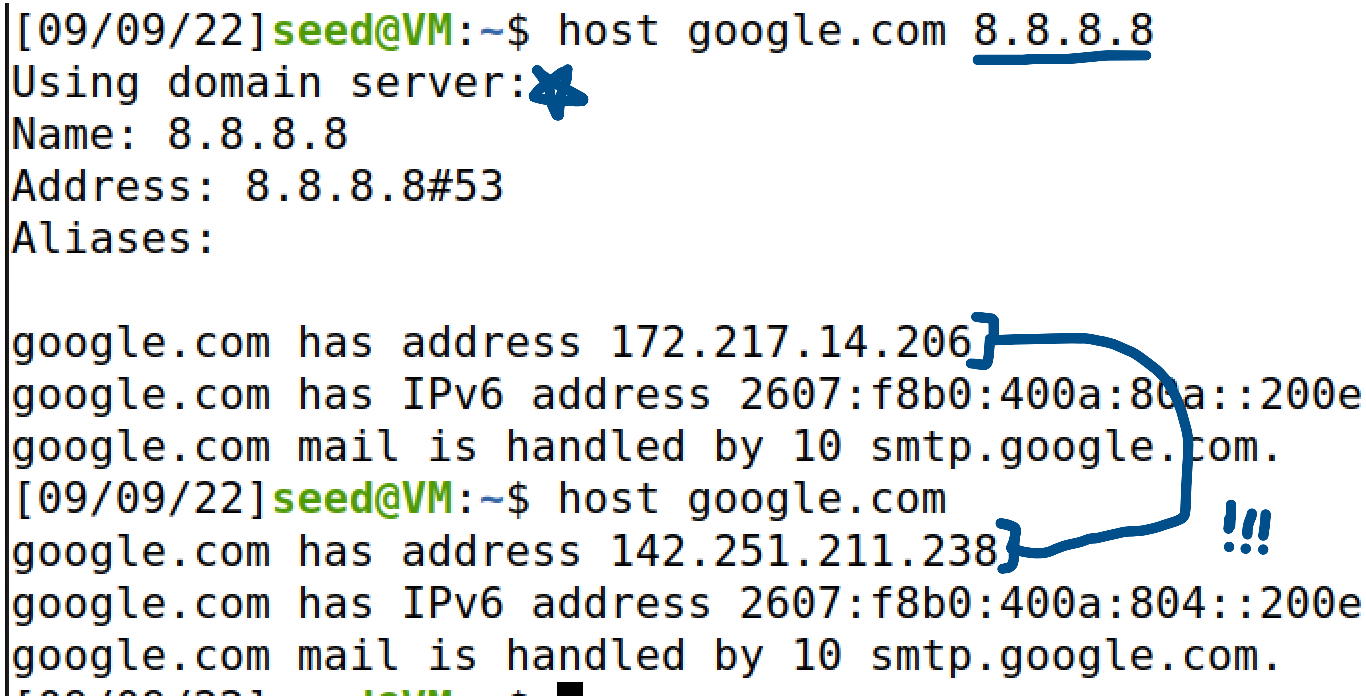
    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

    `host -t MX montana.edu`
  - Load distribution

```
  host huffpost.com | grep "address" | sed -n -e
's/^.*address //p'
```

  - Redirection
    - Look up same host from servers in different regions

    `host google.com 8.8.8.8`



```
[09/09/22]seed@VM:~$ host google.com 8.8.8.8
Using domain server:
Name: 8.8.8.8
Address: 8.8.8.8#53
Aliases:

google.com has address 172.217.14.206
google.com has IPv6 address 2607:f8b0:400a:80a::200e
google.com mail is handled by 10 smtp.google.com.
[09/09/22]seed@VM:~$ host google.com
google.com has address 142.251.211.238
google.com has IPv6 address 2607:f8b0:400a:804::200e
google.com mail is handled by 10 smtp.google.com.
```

DNS Commands

- DNS services
  - Hostname to IP address translation

    `host montana.edu`
  - Hostname to IPv6 address translation
    - `host -t AAAA montana.edu`
  - Host aliasing

    `host -t CNAME img.huffingtonpost.com`
  - Mail server aliasing

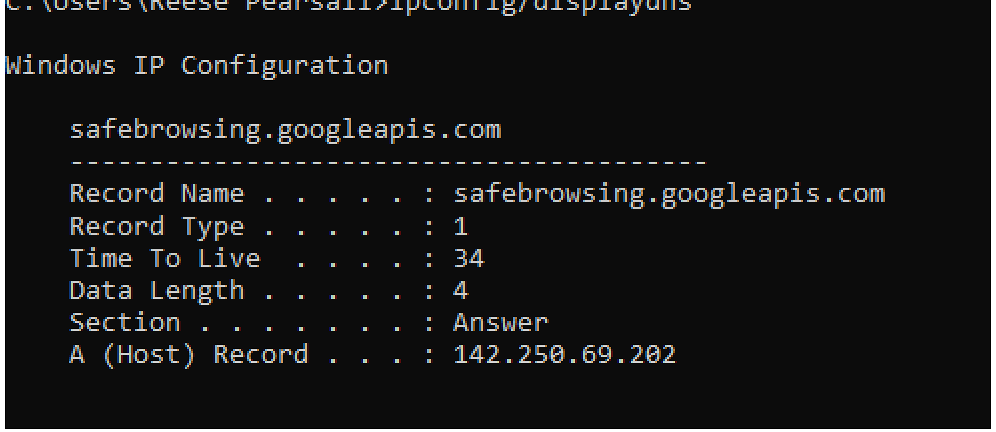
    `host -t MX montana.edu`
  - Load distribution

    `host huffpost.com | grep "address" | sed -n -e 's/^.*address //p'`
  - Redirection
    - Look up same host from servers in different regions

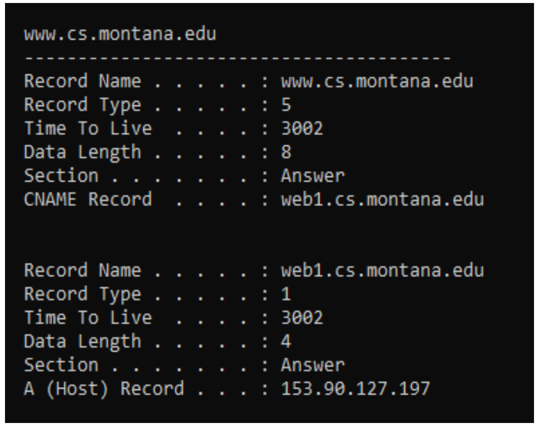
    `host google.com 8.8.8.8`

See cached DNS entries on computer

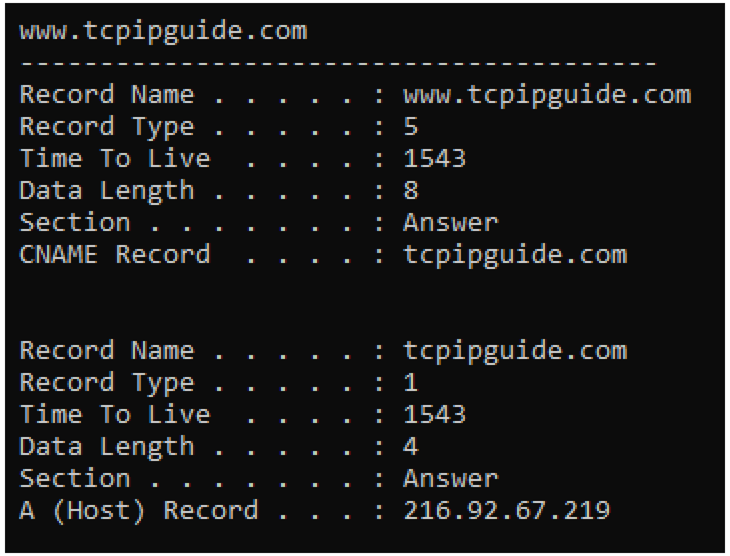
- `ipconfig/displaydns`



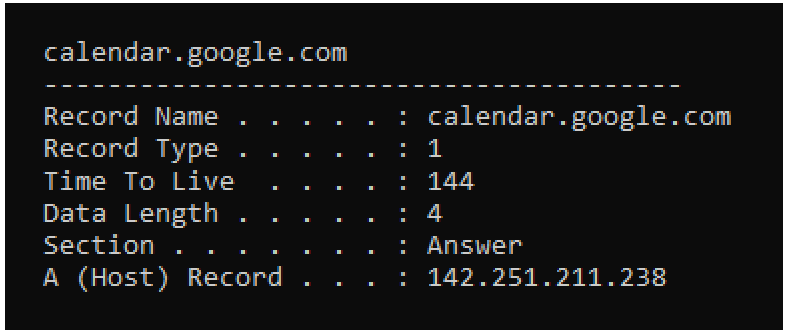
```
C:\Users\Reese Pearsall>ipconfig/displaydns

Windows IP Configuration

    safebrowsing.googleapis.com
    ----------------------------------------
    Record Name . . . . . : safebrowsing.googleapis.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 34
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.250.69.202
```



```
www.cs.montana.edu
----------------------------------------
Record Name . . . . . : www.cs.montana.edu
Record Type . . . . . : 5
Time To Live  . . . . : 3002
Data Length . . . . . : 8
Section . . . . . . . : Answer
CNAME Record  . . . . : web1.cs.montana.edu


Record Name . . . . . : web1.cs.montana.edu
Record Type . . . . . : 1
Time To Live  . . . . : 3002
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 153.90.127.197
```



```
www.tcpipguide.com
----------------------------------------
Record Name . . . . . : www.tcpipguide.com
Record Type . . . . . : 5
Time To Live  . . . . : 1543
Data Length . . . . . : 8
Section . . . . . . . : Answer
CNAME Record  . . . . : tcpipguide.com


Record Name . . . . . : tcpipguide.com
Record Type . . . . . : 1
Time To Live  . . . . : 1543
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 216.92.67.219
```



```
calendar.google.com
----------------------------------------
Record Name . . . . . : calendar.google.com
Record Type . . . . . : 1
Time To Live  . . . . : 144
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 142.251.211.238
```

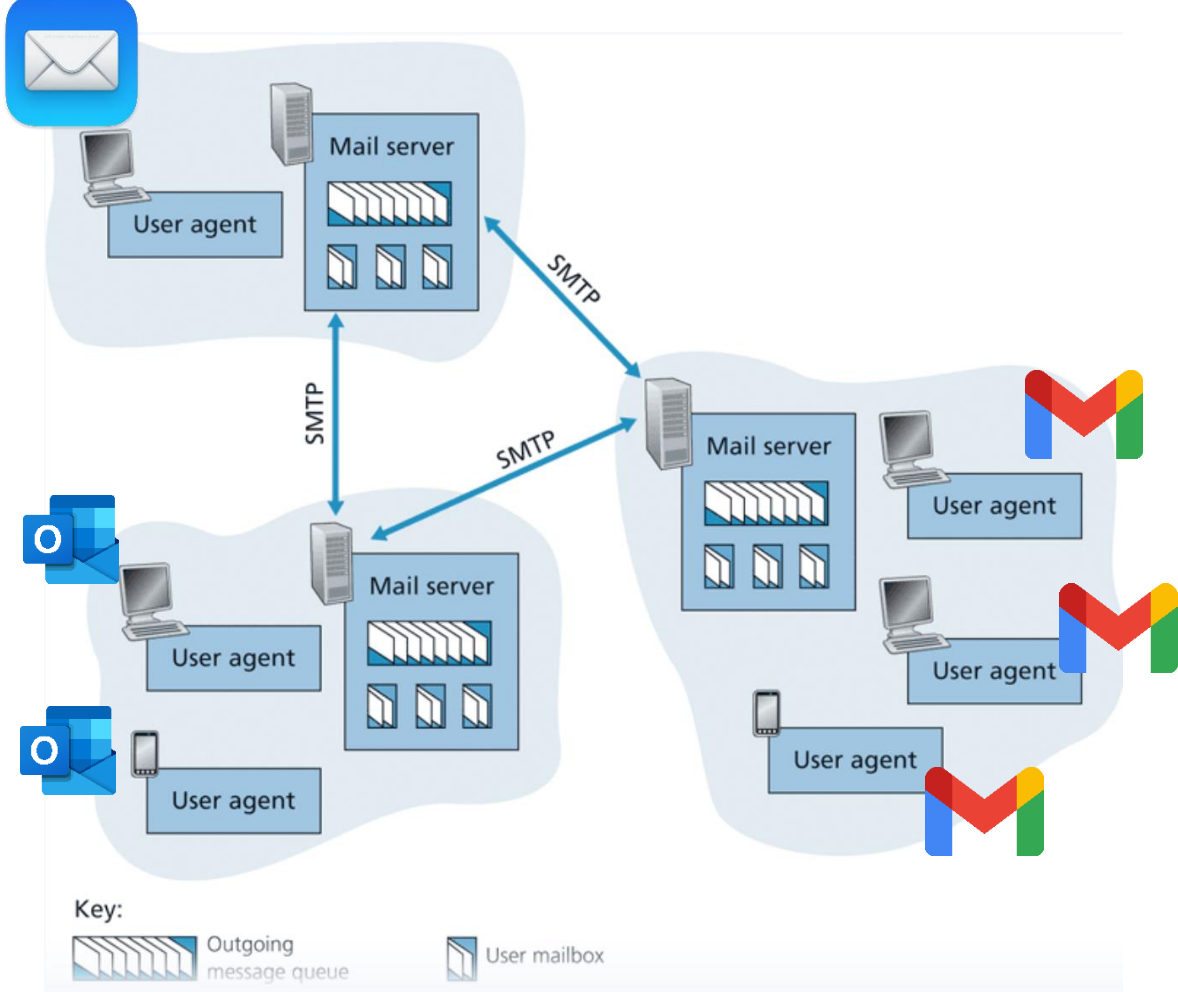
DNS Traffic in Wireshark

SMTP

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

SMTP

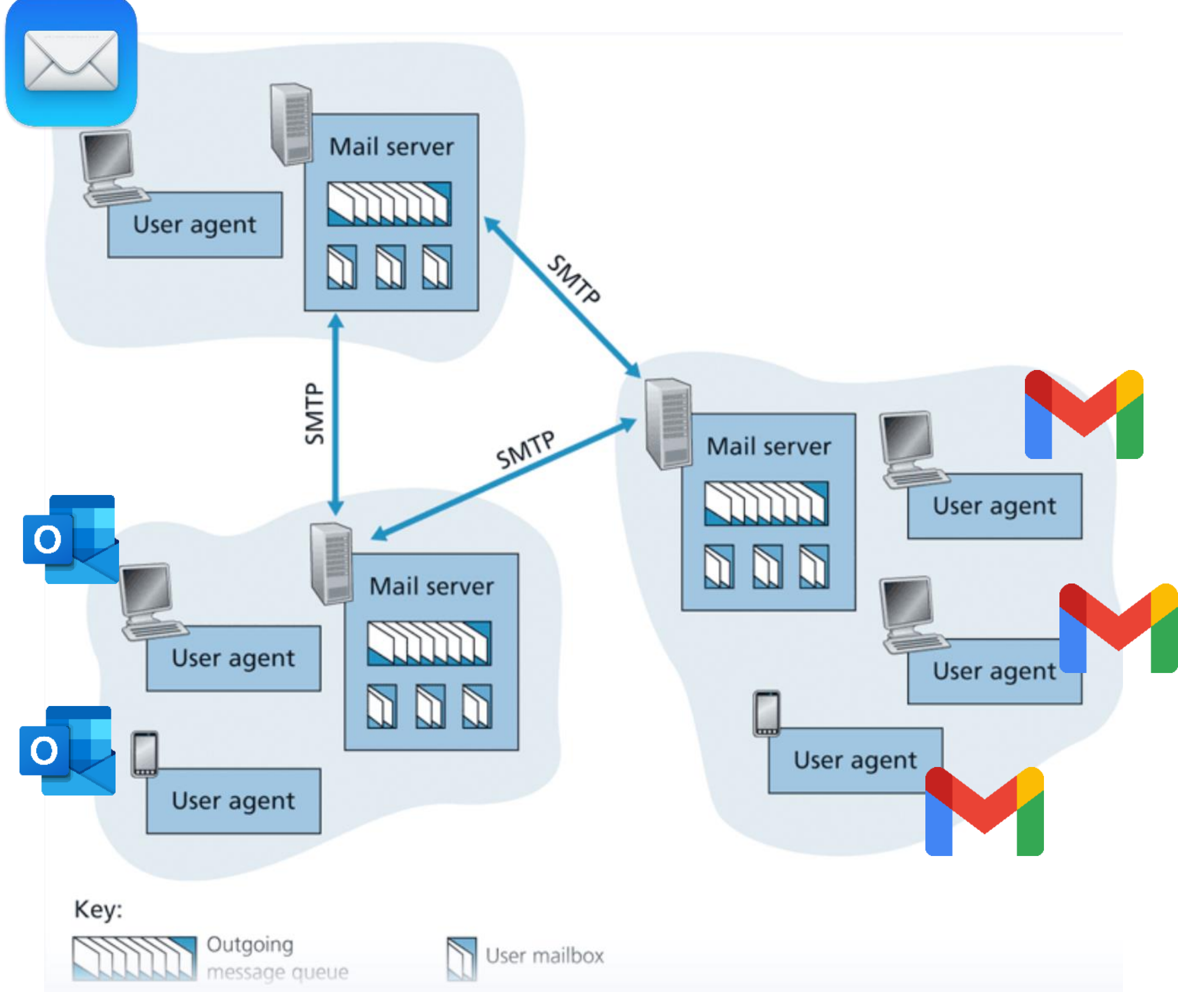
Simple Mail Transfer Protocol (SMTP) is the protocol used for *sending* e-mails from one server to another



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



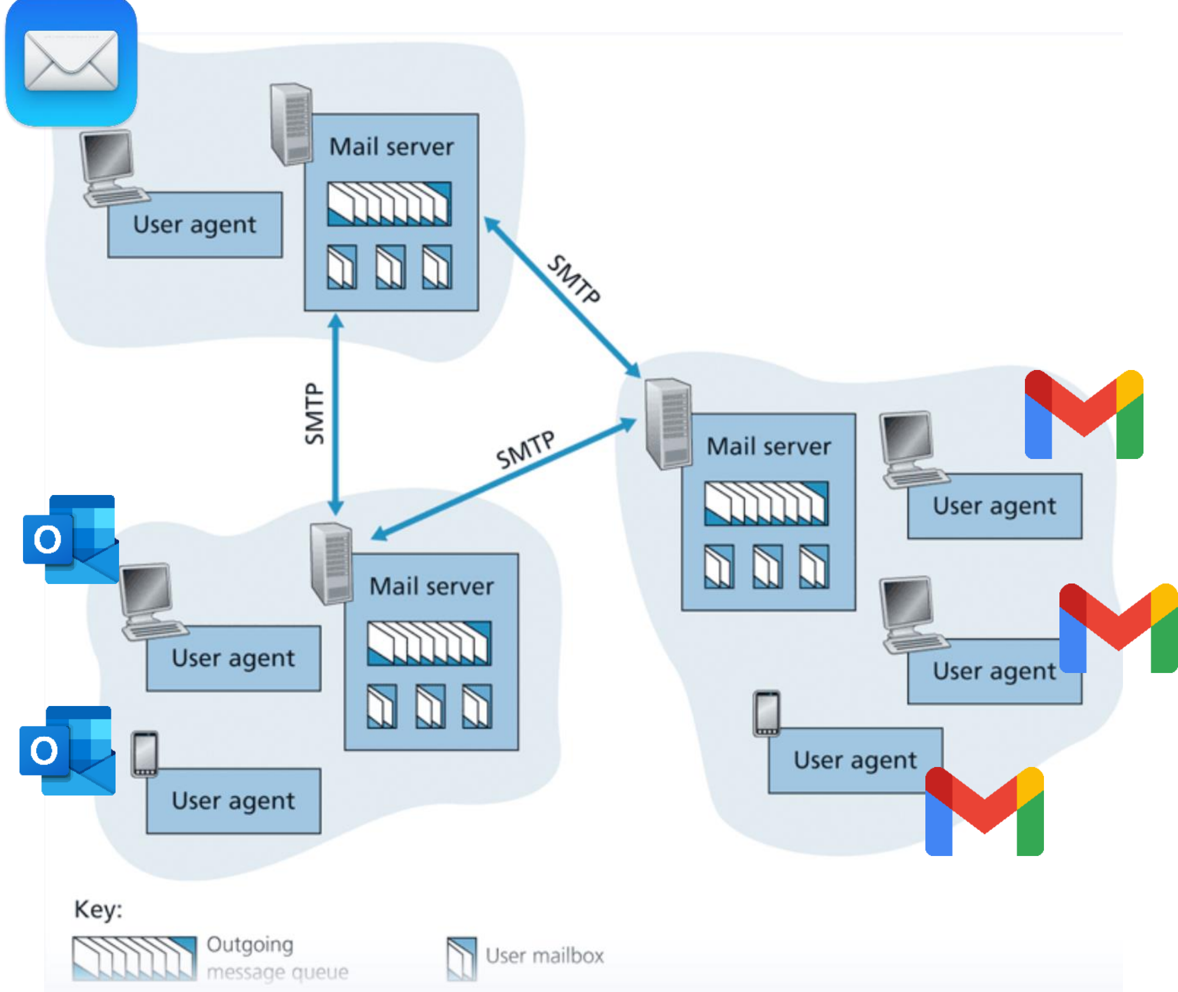
SMTP

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

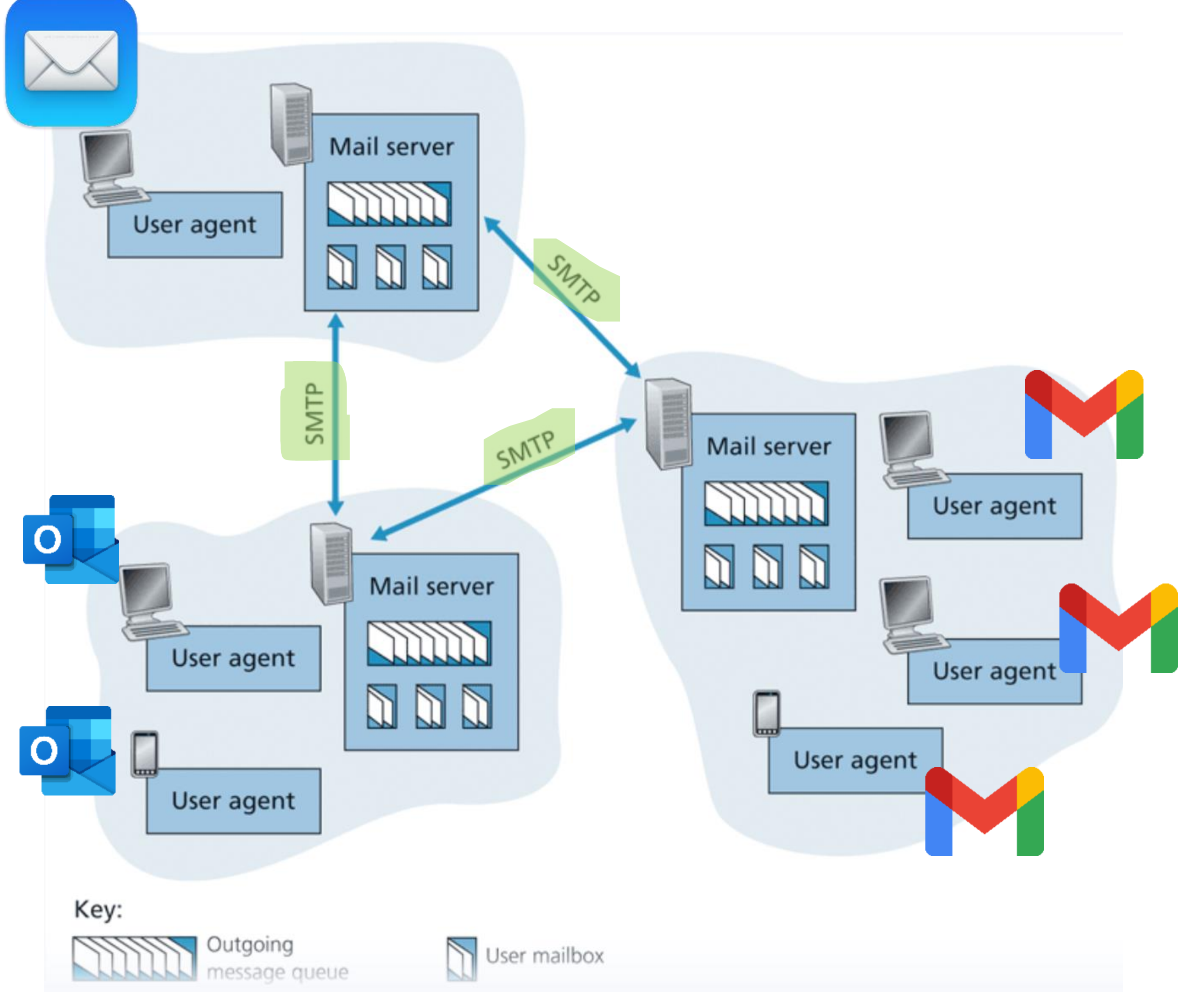
Messages are put in an outgoing **message queue** when they are sent

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

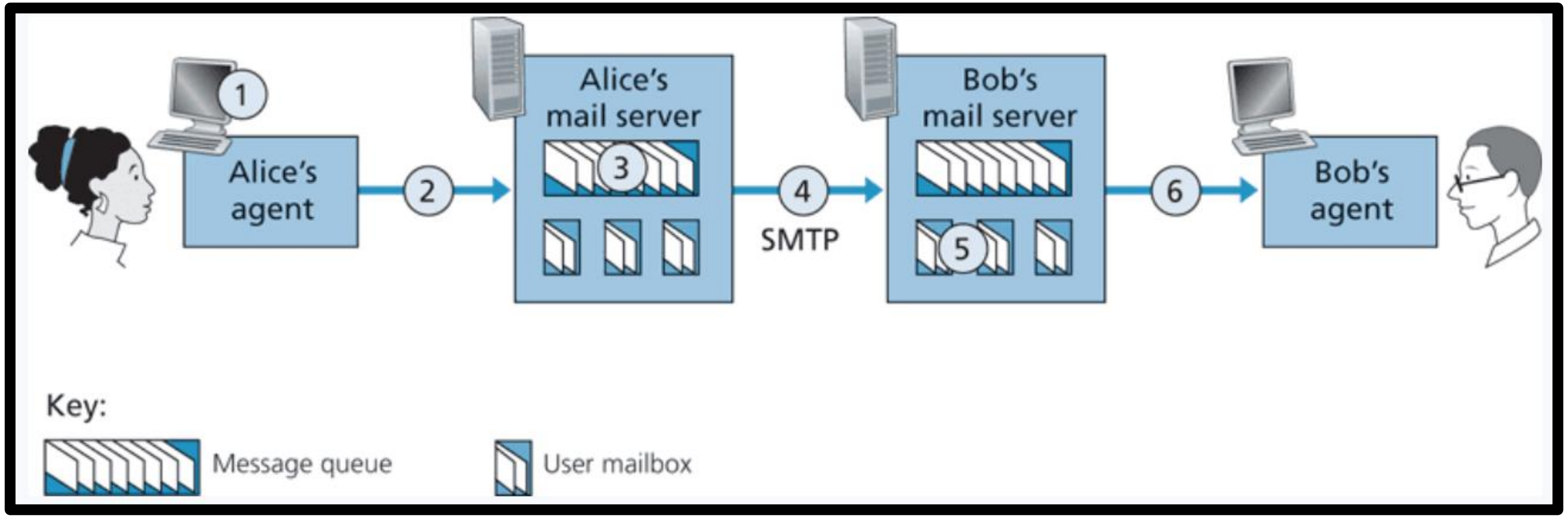


SMTP

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

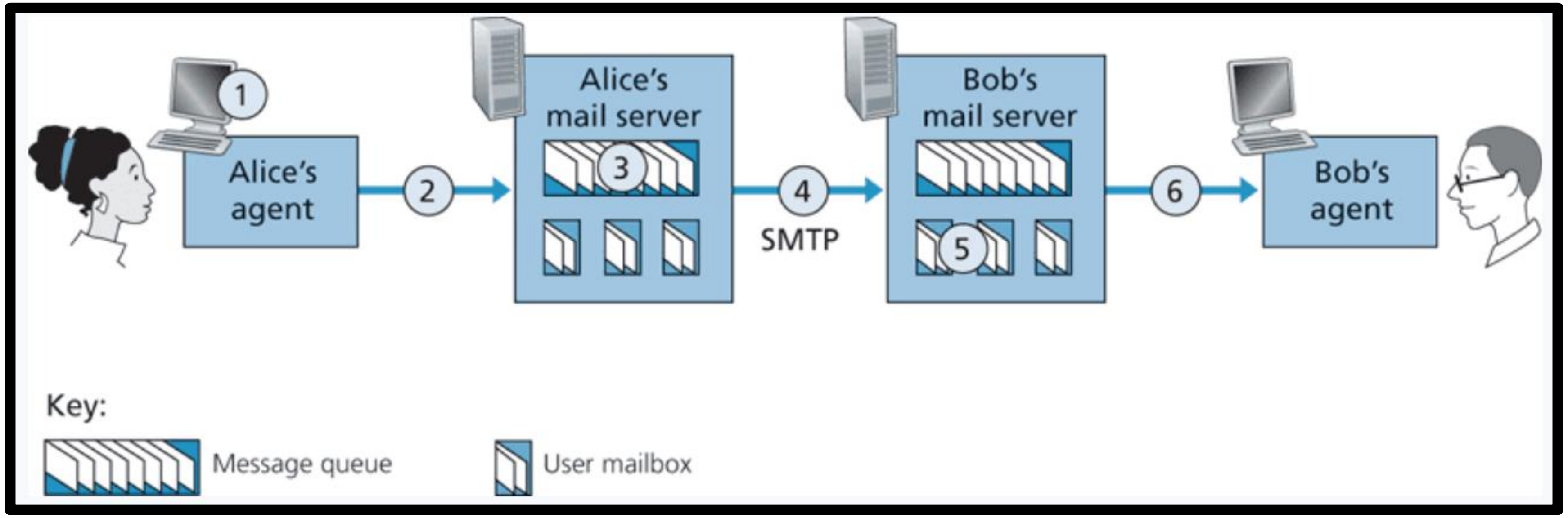


SMTP



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.

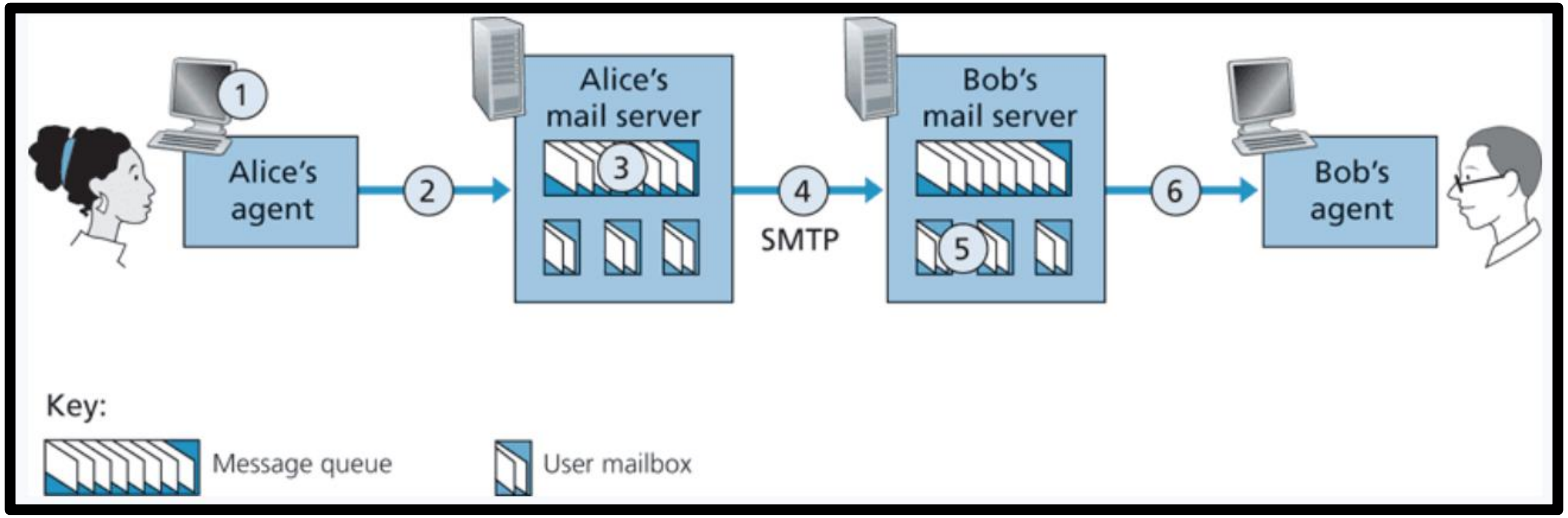
SMTP



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

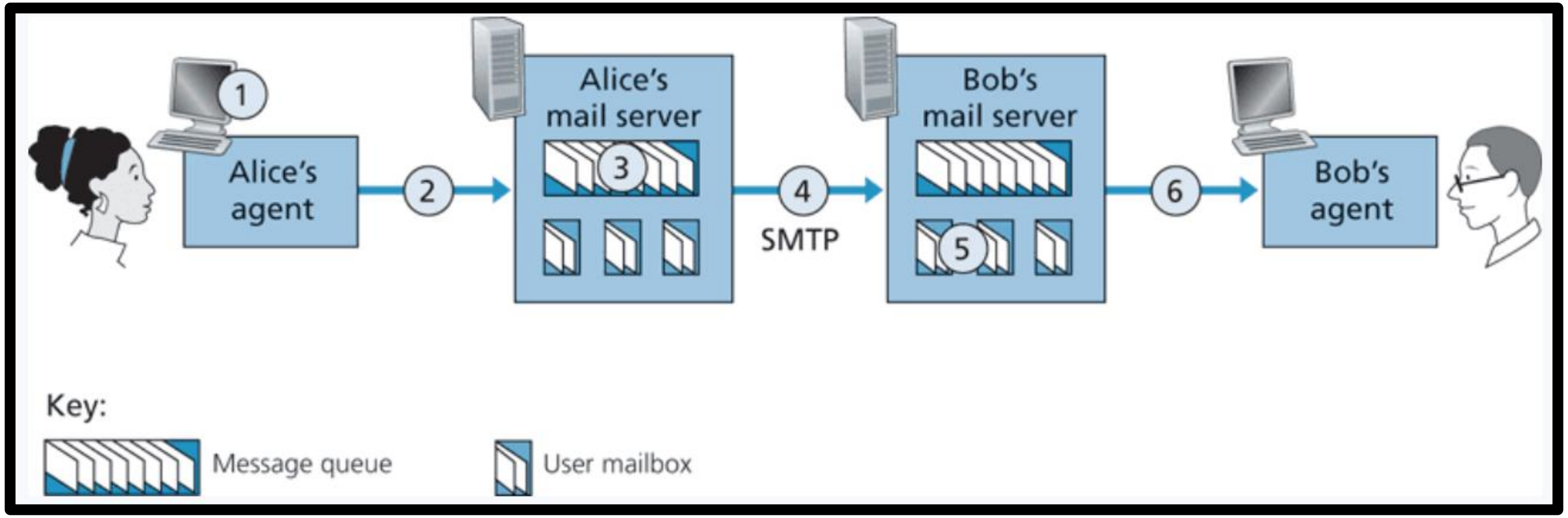


1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.

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



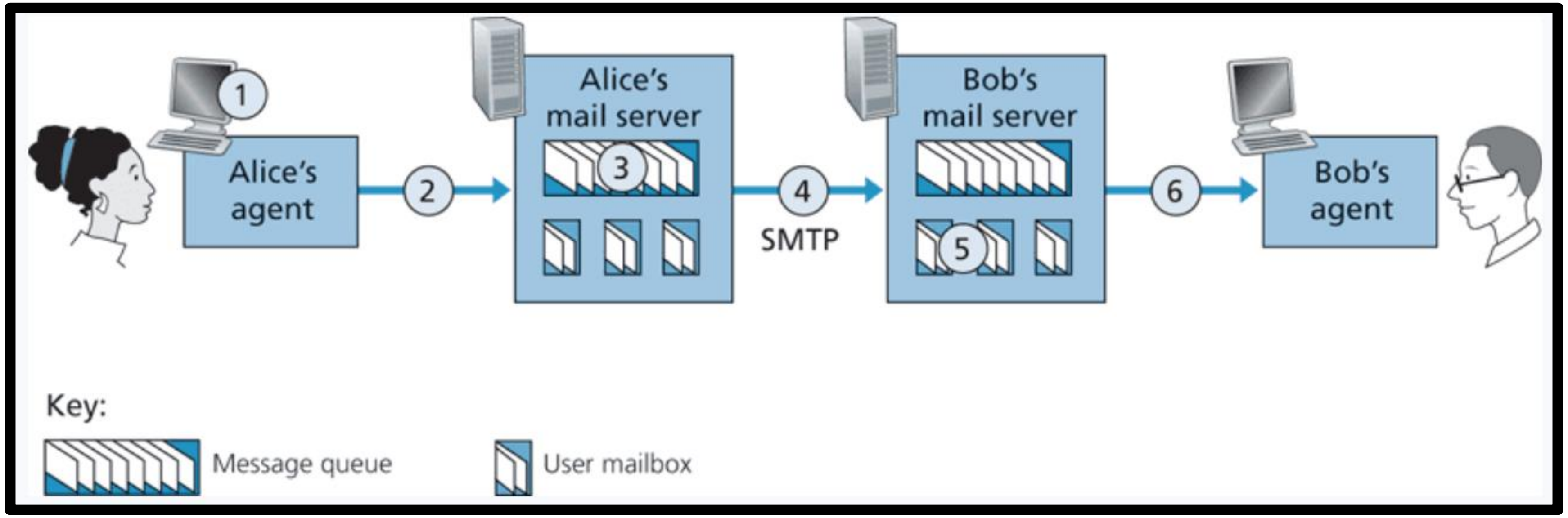
1. Alice invokes her user agent for e-mail, provides Bob's e-mail address (for example, bob@someschool.edu), composes a message, and instructs the user agent to send the message.

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.

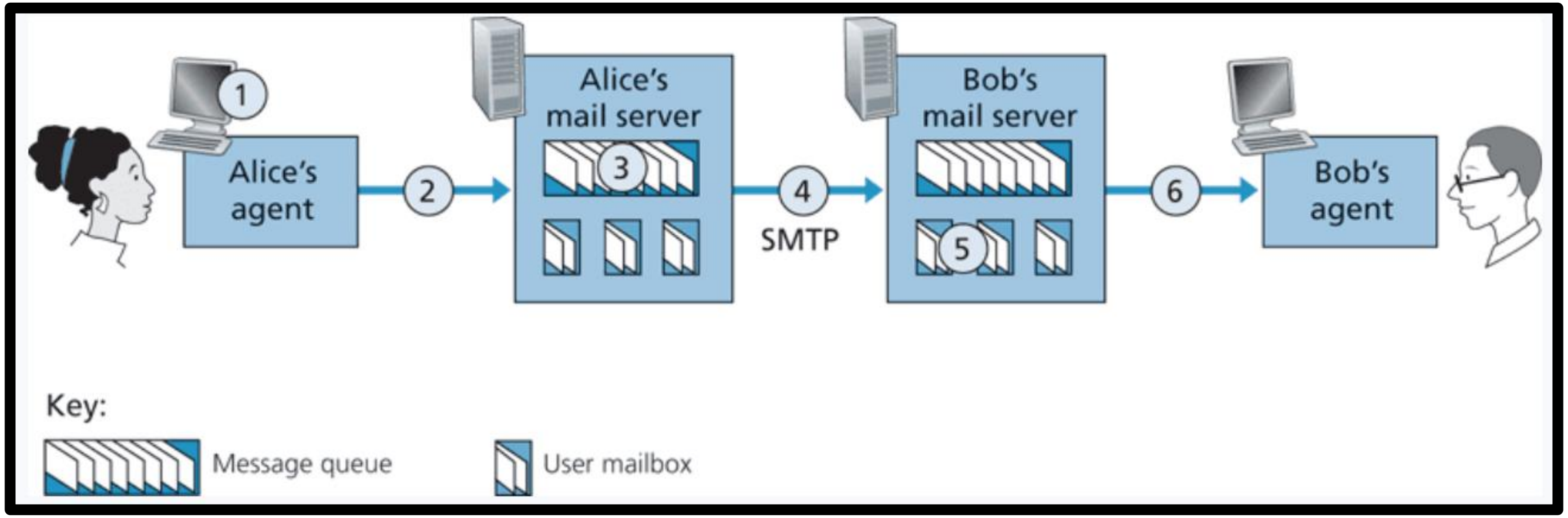
2. Alice's user agent sends the message to her mail server, where it is placed in a message queue.

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.

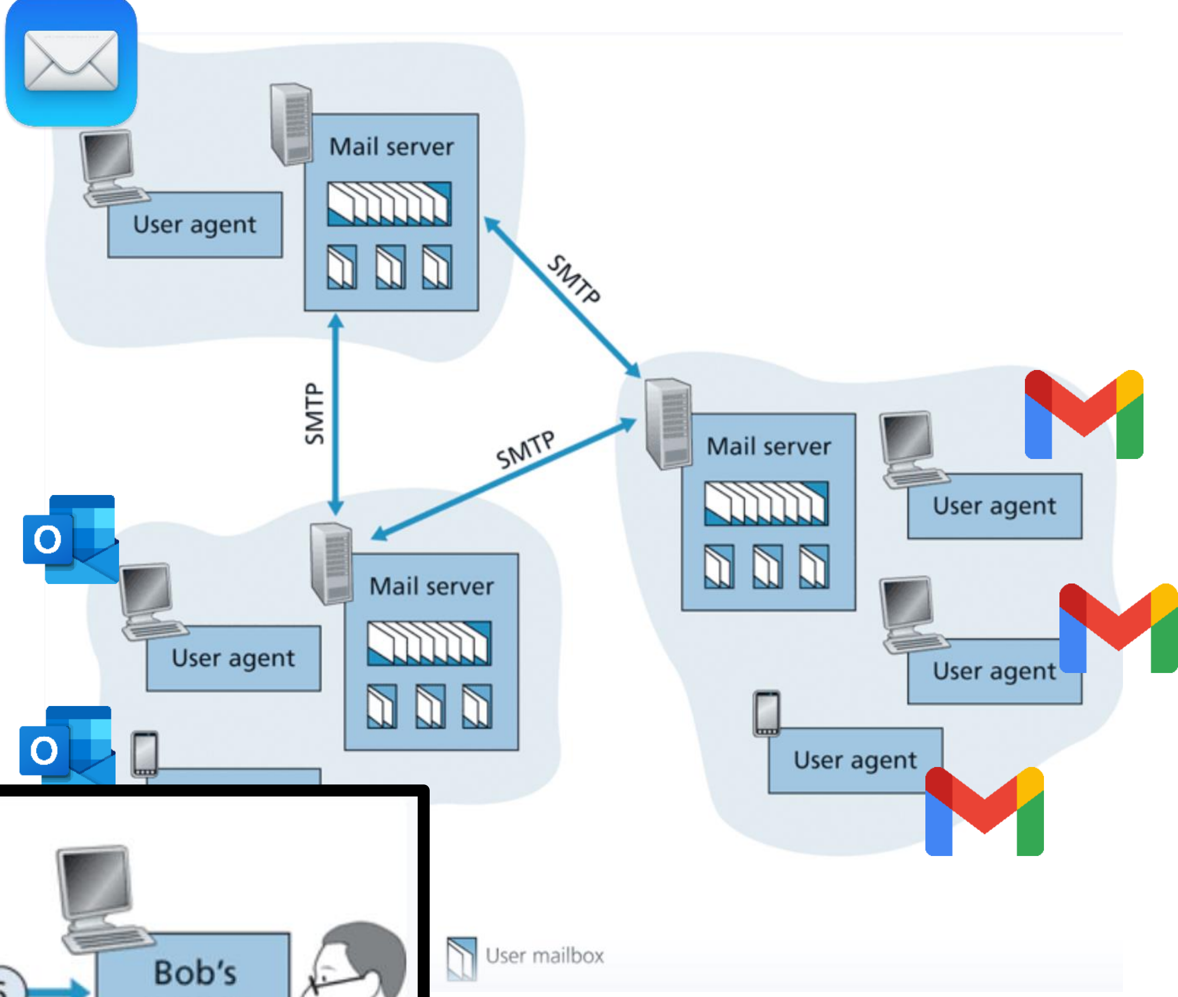
5. At Bob's mail server, the server side of SMTP receives the message. Bob's mail server then places the message in Bob's mailbox.

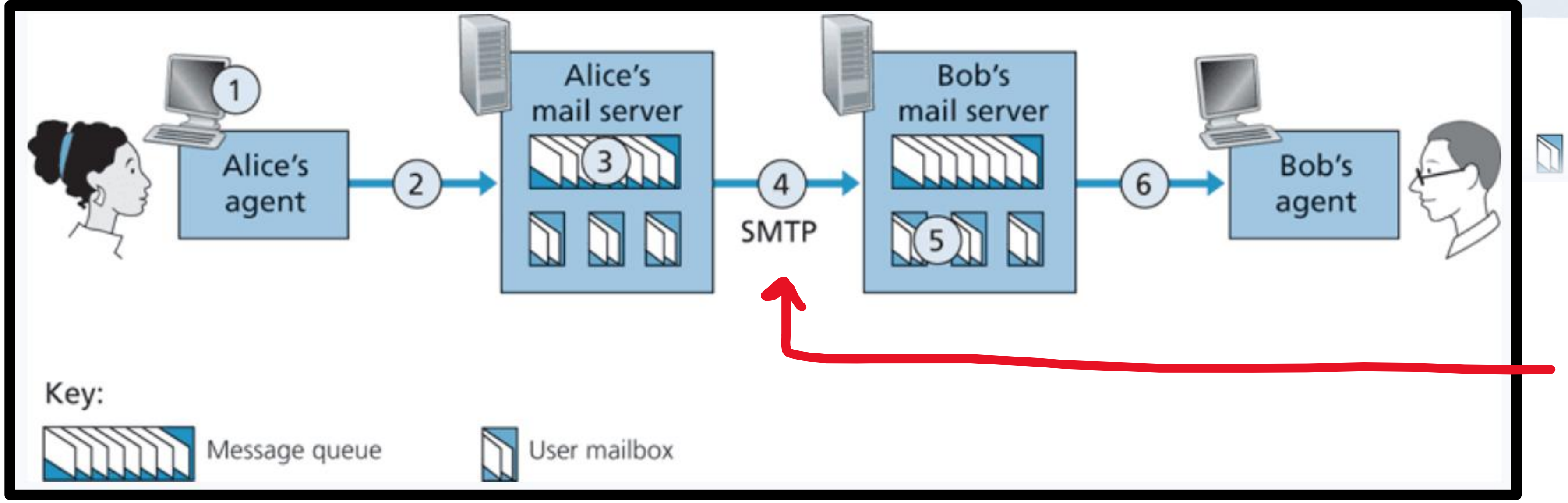
6. Bob invokes his user agent to read the message at his convenience.

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

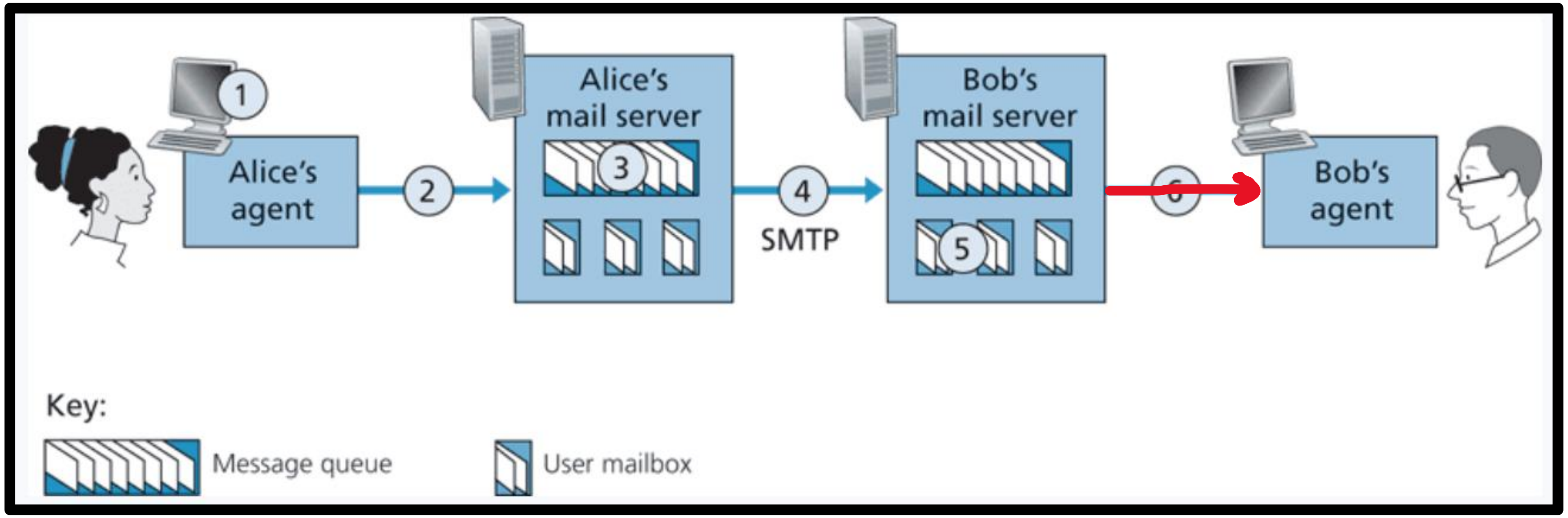




STMP uses TCP for the end-to-end delivery (**DIRECT**)

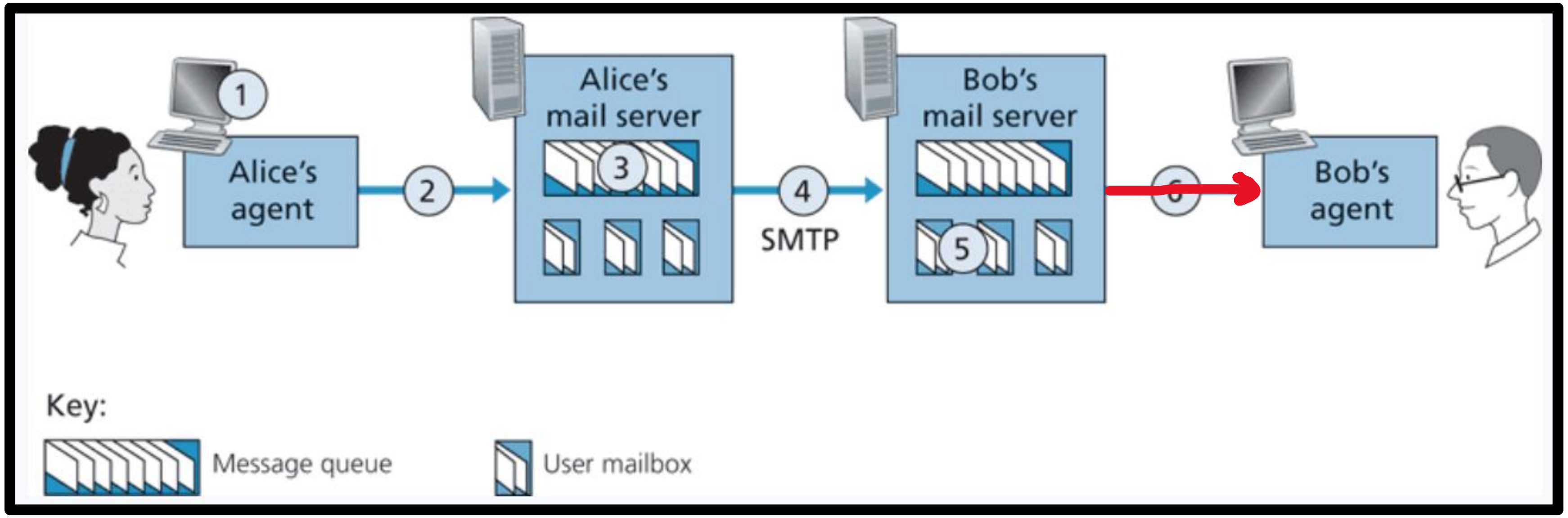
(PORT 25)

SMTP



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

SMTP



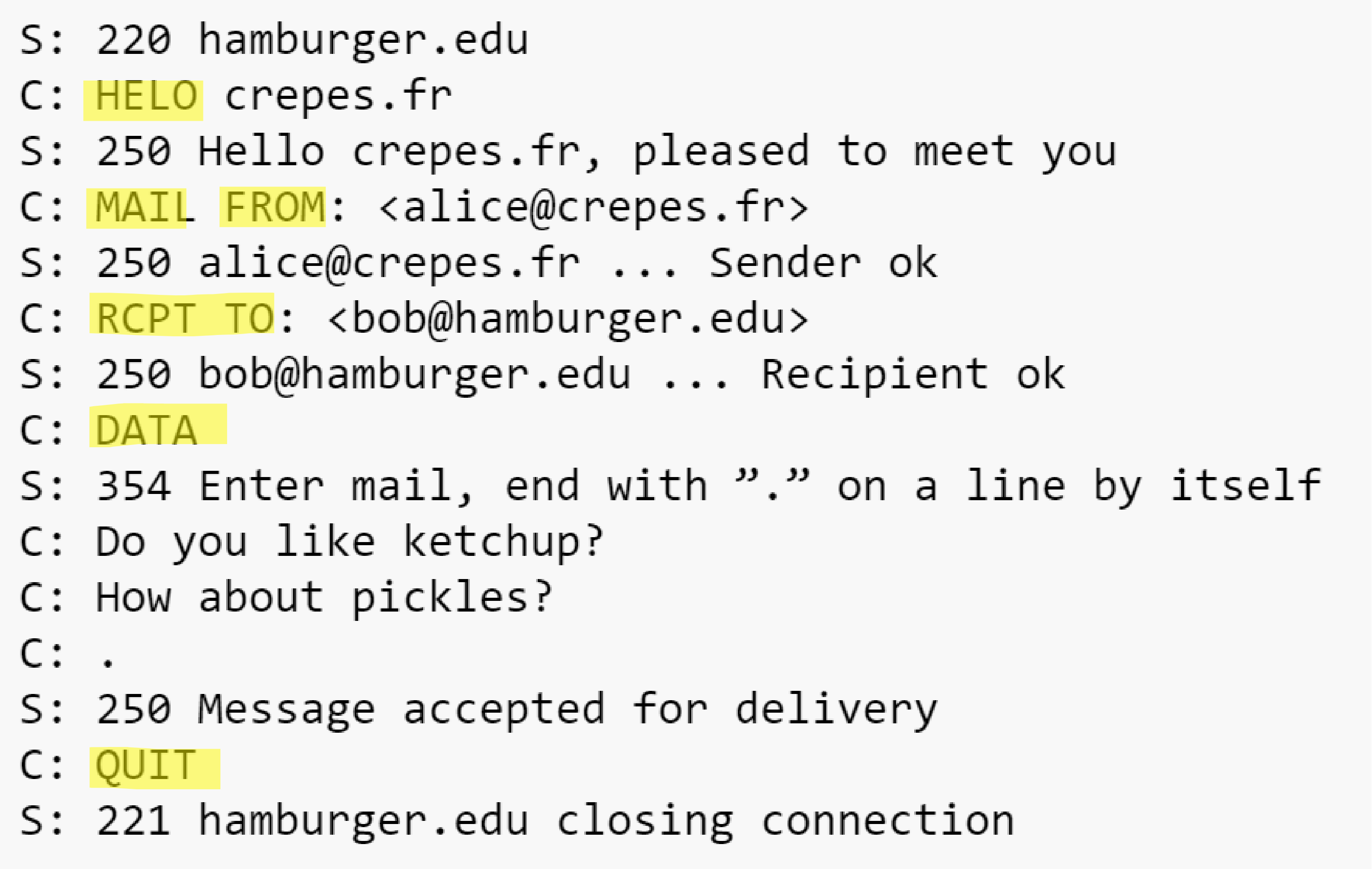
POP3 (post office protocol) or **IMAP** (internet message access protocol) are used to retrieve emails from mail servers.

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

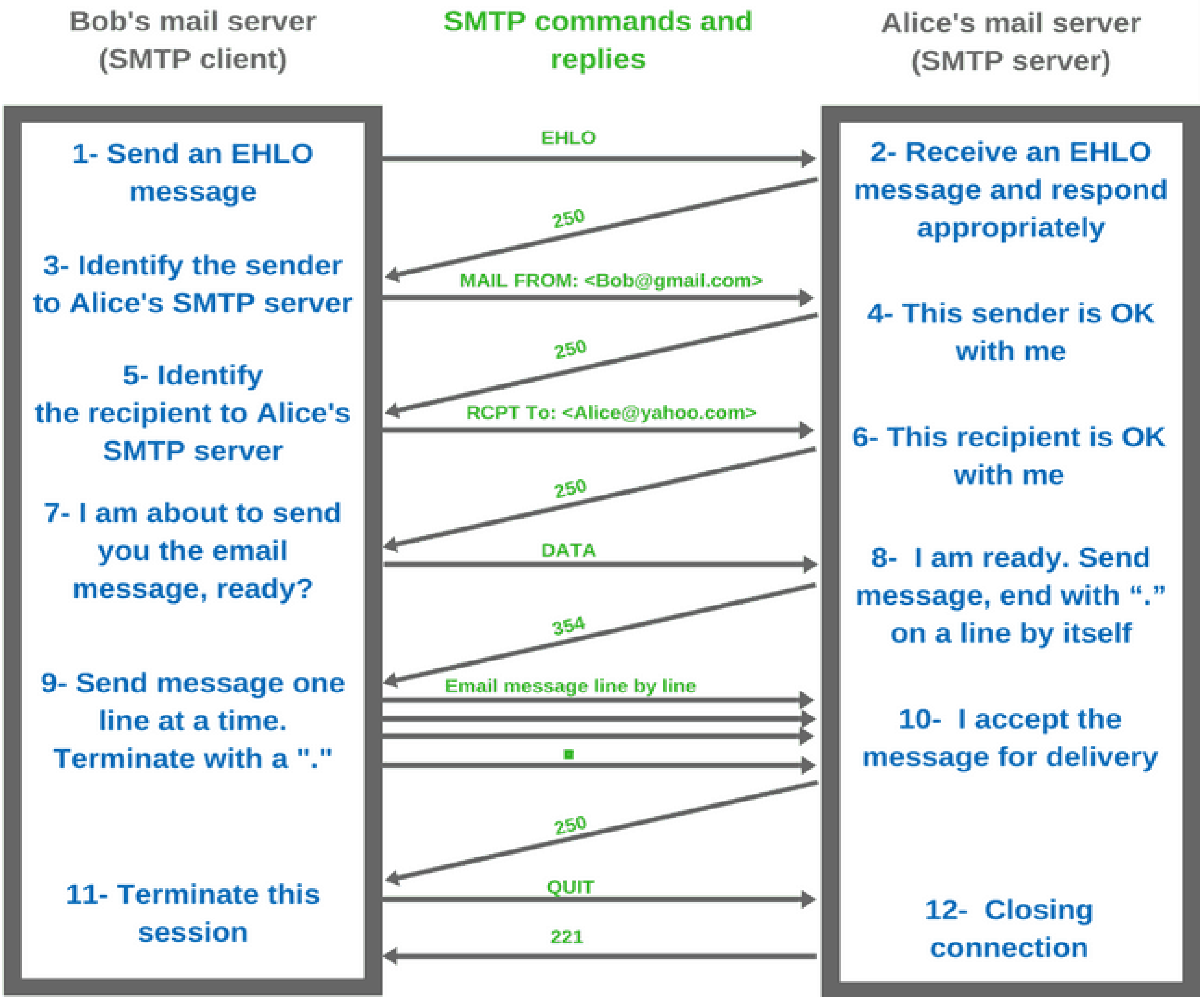
SMTP

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
```


EHLO
250
MAIL FROM: <Bob@gmail.com>
250
RCPT To: <Alice@yahoo.com>
250
DATA
354
Email message line by line
.
250
QUIT
221

Bob's mail server
(SMTP client)

SMTP commands and replies

Alice's mail server
(SMTP server)

1- Send an EHLO message

2- Receive an EHLO message and respond appropriately

3- Identify the sender to Alice's SMTP server

4- This sender is OK with me

5- Identify the recipient to Alice's SMTP server

6- This recipient is OK with me

7- I am about to send you the email message, ready?

8- I am ready. Send message, end with "." on a line by itself

9- Send message one line at a time. Terminate with a "."

10- I accept the message for delivery

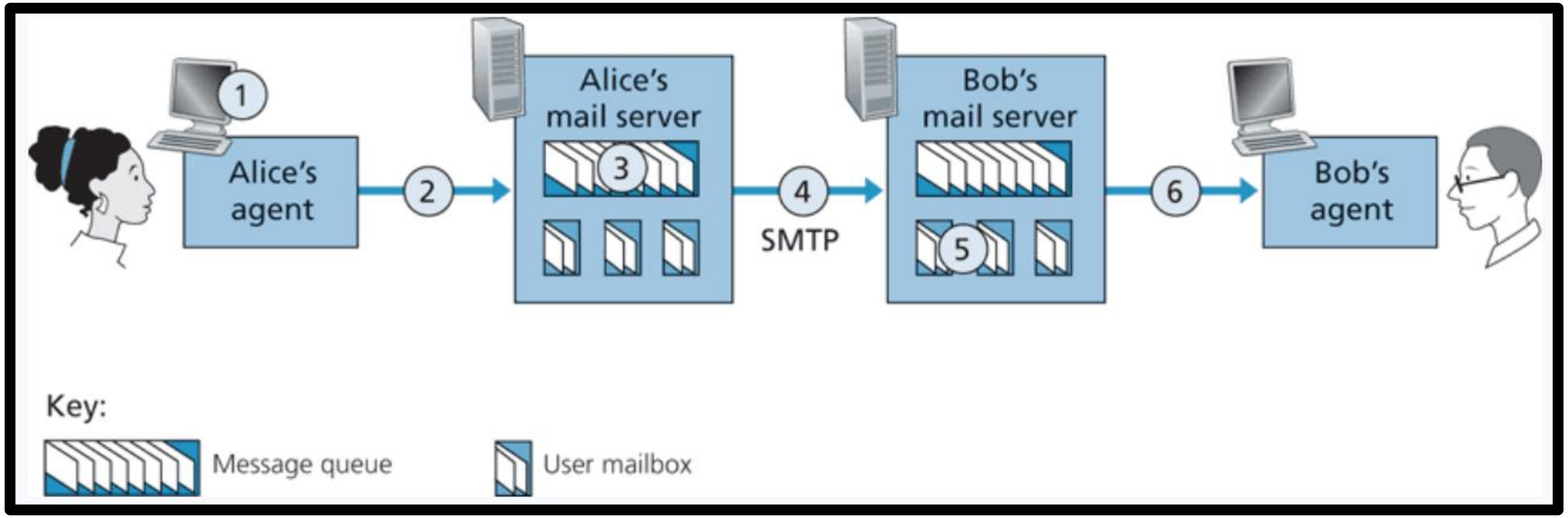
11- Terminate this session

12- Closing connection

SMTP

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

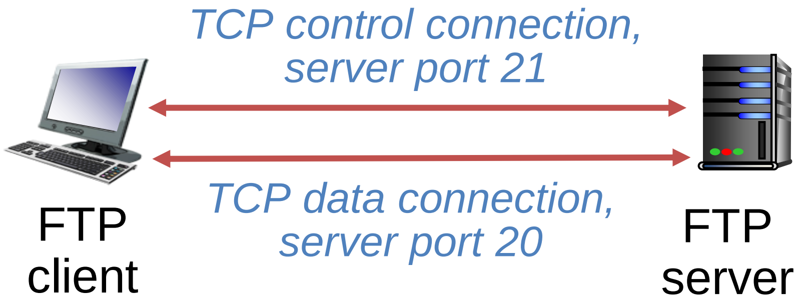
Ports 25 is reserved for SMTP traffic (and also port 587 & 465)



SMTP Traffic in Wireshark

FTP

File Transfer Protocol (FTP)- protocol used for transferring files from server to client



TCP control connection, server port 21
TCP data connection, server port 20
FTP client
FTP server

- FTP communicates over two connections
  - Port 21 for control information
  - Port 20 for data

- Differences from HTTP
  - Control communication “out-of-band”
  - Server maintains per client state: authentication, current directory

- **FTP procedure:**
  1. FTP client contacts FTP server at port 21, using TCP
  2. Client authorized over control connection
  3. Client browses remote directory, sends commands over control connection
  4. When server receives file transfer command, server opens 2nd TCP data connection (for file) to client
  5. After transferring one file, server closes data connection

Why use a separate control connection?

WINSCP