Master of Science in
Genomics and Bioinformatics

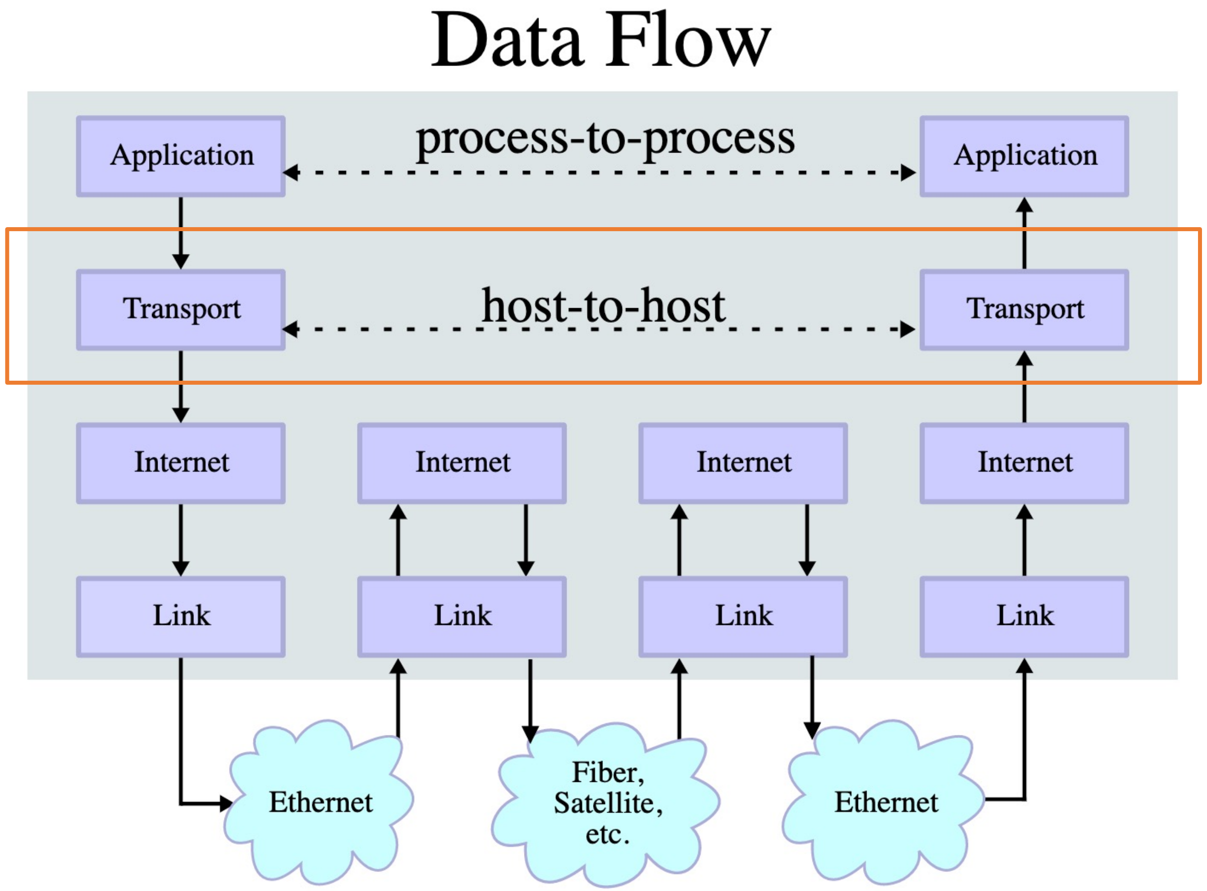
Lecture 10-1
Python Network Programing

GNBF5010

2021R1

Transmission Control Protocol (TCP)

- A standard that enables applications and computing devices to exchange messages over a network
- Designed to send packets across the internet and ensure the successful delivery of data

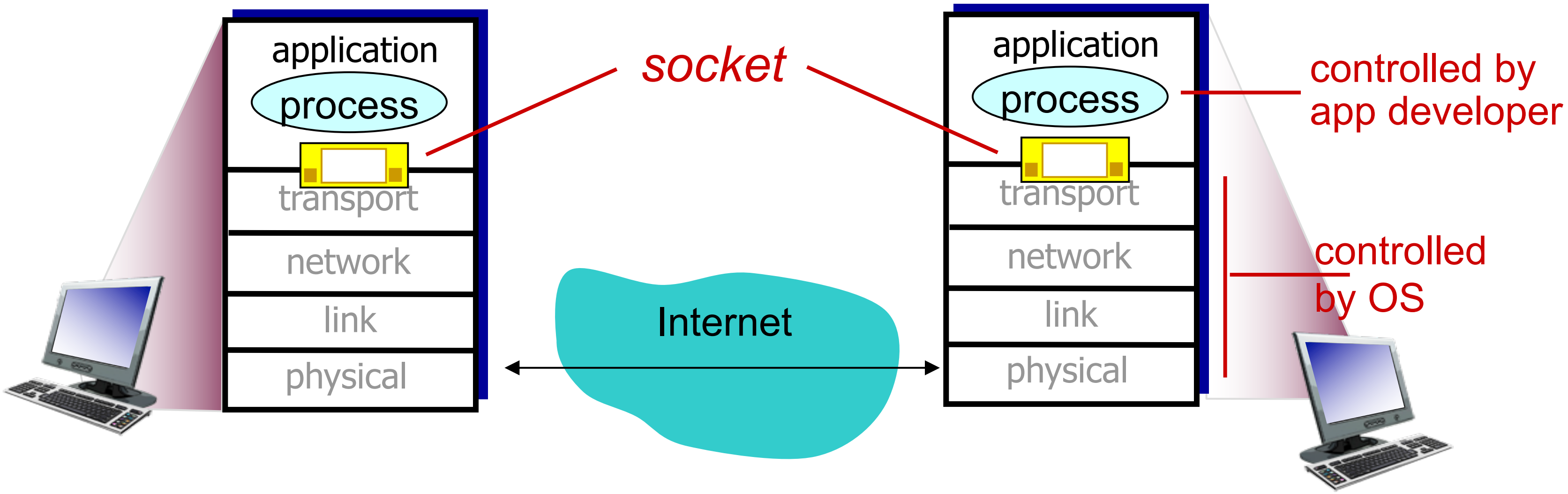


https://www.fortinet.com/resources/cyberglossary/tcp-ip

https://en.wikipedia.org/wiki/Internet_protocol_suite

TCP Connections / Sockets

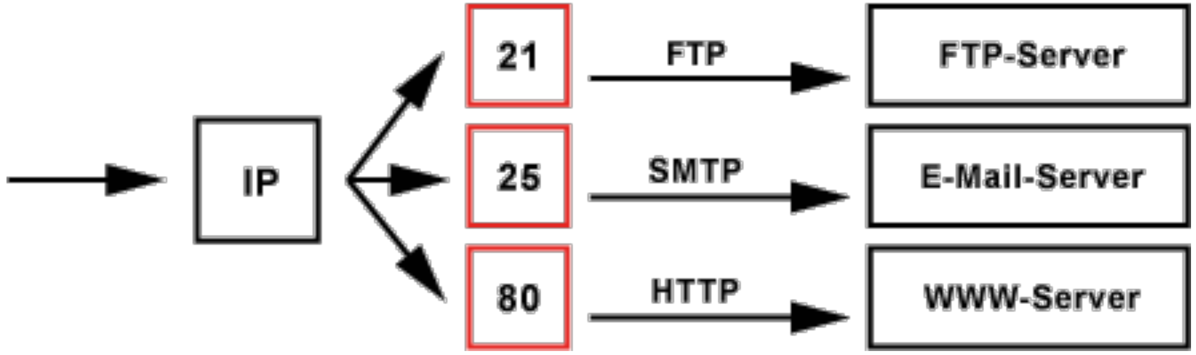
- A socket is one endpoint of a two-way communication link between two programs running on the network.
- A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.



Source: Computer Networking: A Top-Down Approach, 7th edition. Jim Kurose, Keith Ross. Pearson/Addison Wesley 2016.
https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html

Examples of standard TCP ports

IP data packet TCP-port Protocol Application



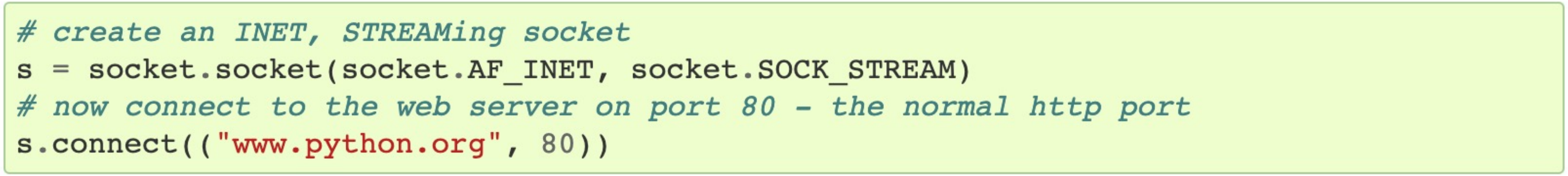
(Server side)

Port number	Protocol	Use
20/21	FTP	File transfer (FTP server)
22	FTPS/SSH	SSH connection (SSH server)
23	Telnet	Console (server)
25	SMTP	Outbox (SMTP server)
80	HTTP	World Wide Web (web server)
110	POP3	Inbox (POP server)

https://www.elektronik-kompendium.de/sites/net/1812041.htm

Creating Sockets with Python

- When you click on a link, your browser does something like the following:



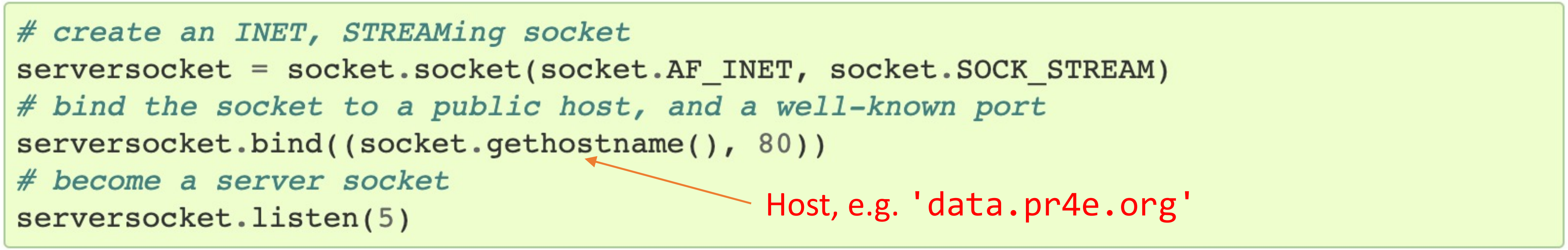
```
# create an INET, STREAMing socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# now connect to the web server on port 80 - the normal http port
s.connect(("www.python.org", 80))
```

- When the connect completes, the socket s can be used to send in a request for the text of the page.
- The same socket will read the reply, and then be destroyed.
- This is on client side.

https://docs.python.org/3/howto/sockets.html

Creating Sockets with Python

- What happens in the web server is a bit more complex.
- First, the web server creates a "server socket":



```
# create an INET, STREAMing socket
serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# bind the socket to a public host, and a well-known port
serversocket.bind((socket.gethostname(), 80))
# become a server socket
serversocket.listen(5)
```

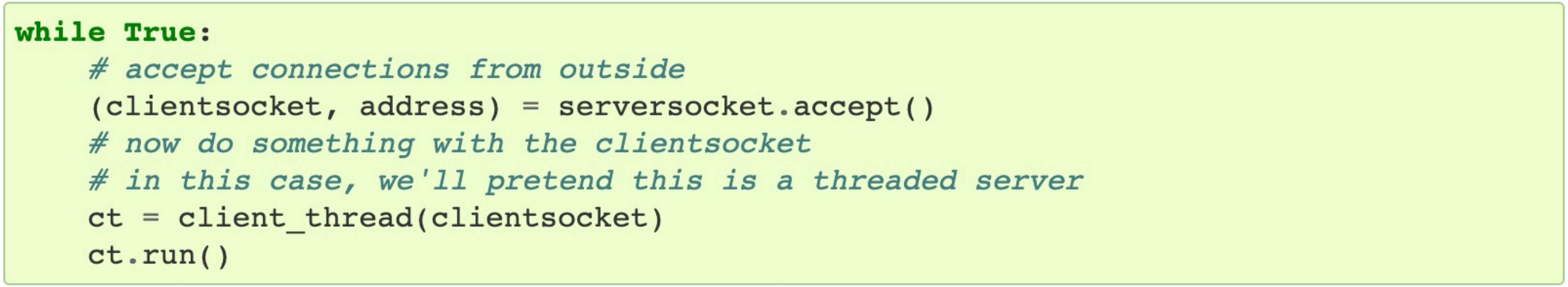
Host, e.g. `'data.pr4e.org'`

- If we use `s.bind(('localhost', 80))` or `s.bind(('127.0.0.1', 80))`, we will still have a "server" socket, but one that was only visible within the same machine.

https://docs.python.org/3/howto/sockets.html

Creating Sockets with Python

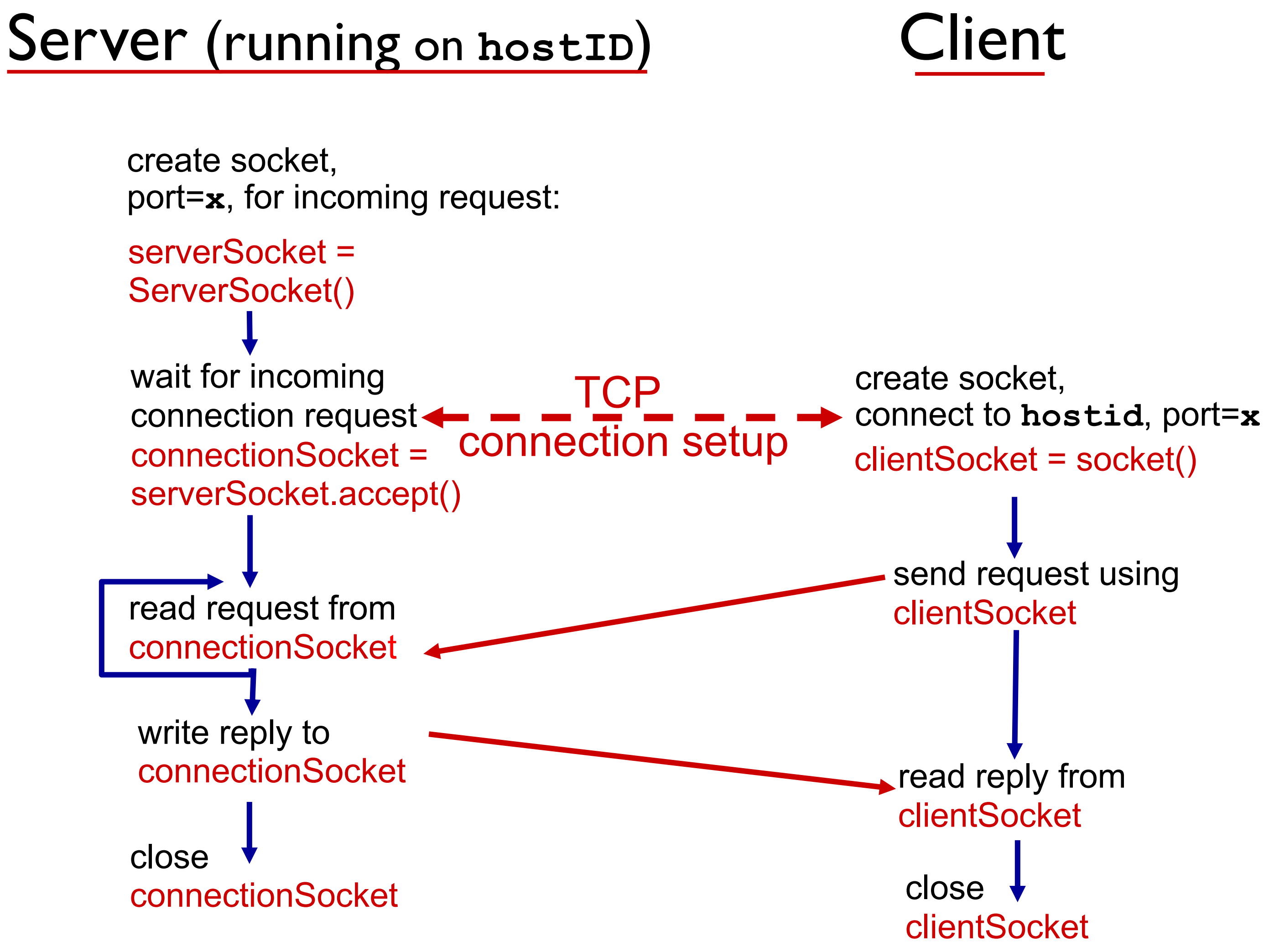
- Now that we have a “server” socket, listening on port 80, we can enter the main loop of the web server:



```
while True:
    # accept connections from outside
    (clientsocket, address) = serversocket.accept()
    # now do something with the clientsocket
    # in this case, we'll pretend this is a threaded server
    ct = client_thread(clientsocket)
    ct.run()
```

https://docs.python.org/3/howto/sockets.html

Socket interaction between server and client



Server (running on hostID)

Client

create socket,
port=x, for incoming request:

serverSocket = ServerSocket()

wait for incoming connection request

connectionSocket = serverSocket.accept()

TCP connection setup

create socket,
connect to hostid, port=x

clientSocket = socket()

send request using clientSocket

read request from connectionSocket

write reply to connectionSocket

read reply from clientSocket

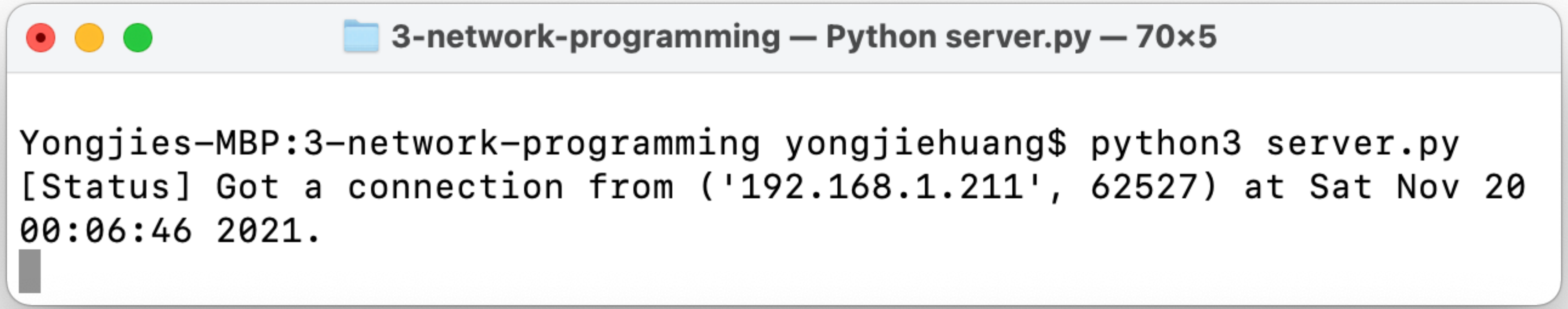
close connectionSocket

close clientSocket

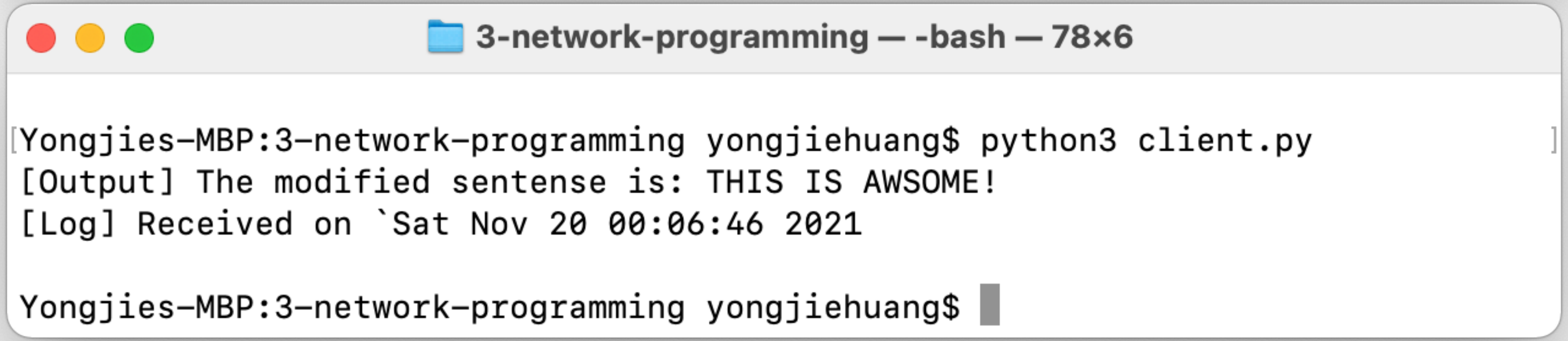
Source: Computer Networking: A Top-Down Approach, 7th edition. Jim Kurose, Keith Ross. Pearson/Addison Wesley 2016.

A simple example

- See server.py and client.py
- Run the server.py on a terminal; then run client.py on another terminal



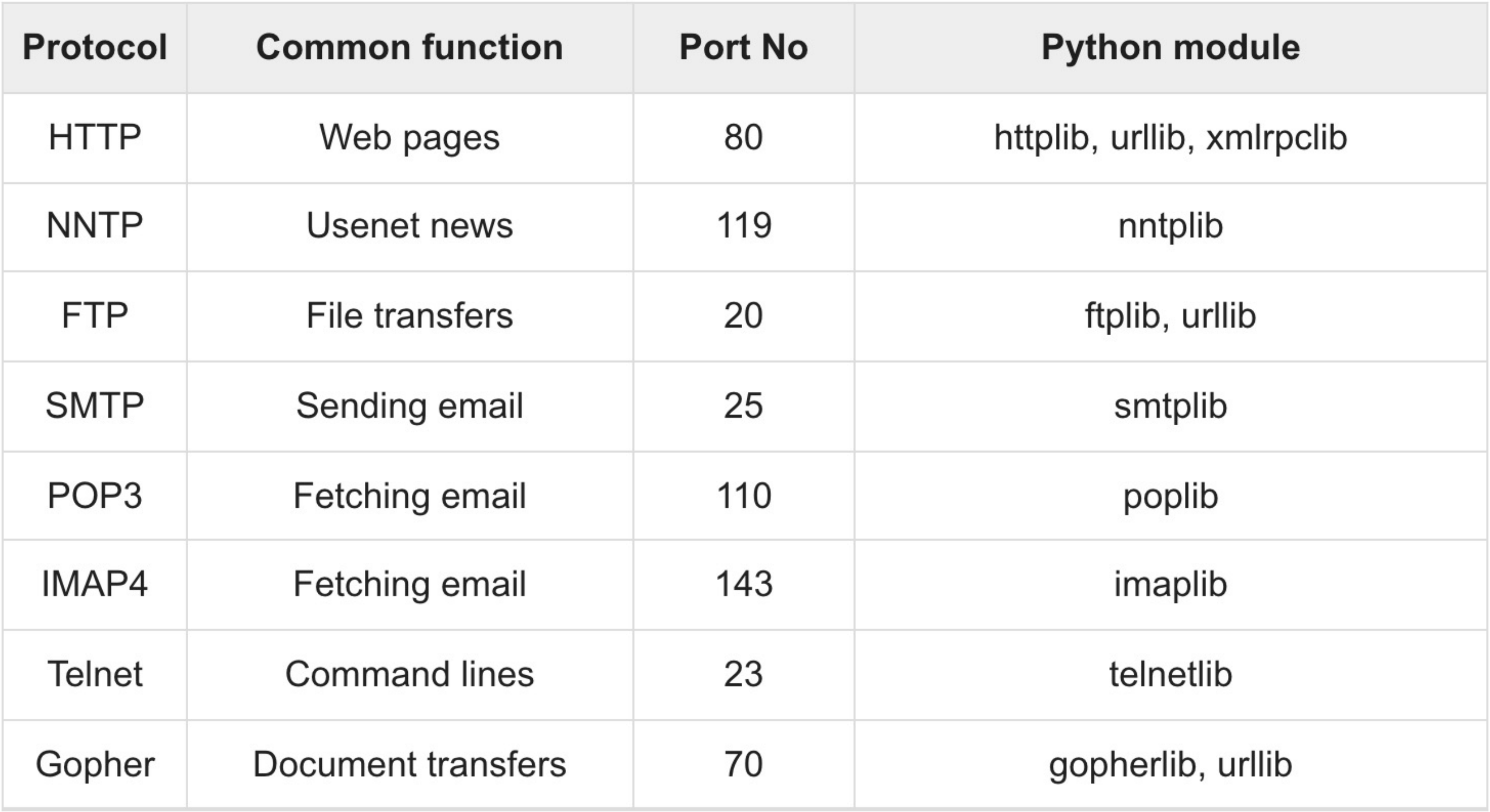
```
Yongjies-MBP:3-network-programming yongjiehuang$ python3 server.py
[Status] Got a connection from ('192.168.1.211', 62527) at Sat Nov 20
00:06:46 2021.
```



```
Yongjies-MBP:3-network-programming yongjiehuang$ python3 client.py
[Output] The modified sentense is: THIS IS AWSOME!
[Log] Received on `Sat Nov 20 00:06:46 2021

Yongjies-MBP:3-network-programming yongjiehuang$
```

Python network/internet modules



Protocol	Common function	Port No	Python module
HTTP	Web pages	80	httplib, urllib, xmlrpclib
NNTP	Usenet news	119	nntplib
FTP	File transfers	20	ftplib, urllib
SMTP	Sending email	25	smtplib
POP3	Fetching email	110	poplib
IMAP4	Fetching email	143	imaplib
Telnet	Command lines	23	telnetlib
Gopher	Document transfers	70	gopherlib, urllib

References

- Chapter 12 Networked programs - from Python for Everybody
  - https://www.py4e.com/html3/12-network

- Network Programming - from Tutorials Point
  - https://www.tutorialspoint.com/python/python_networking.htm