

CS168: Discussion 3

...

Intro to the Internet
Fall 2022

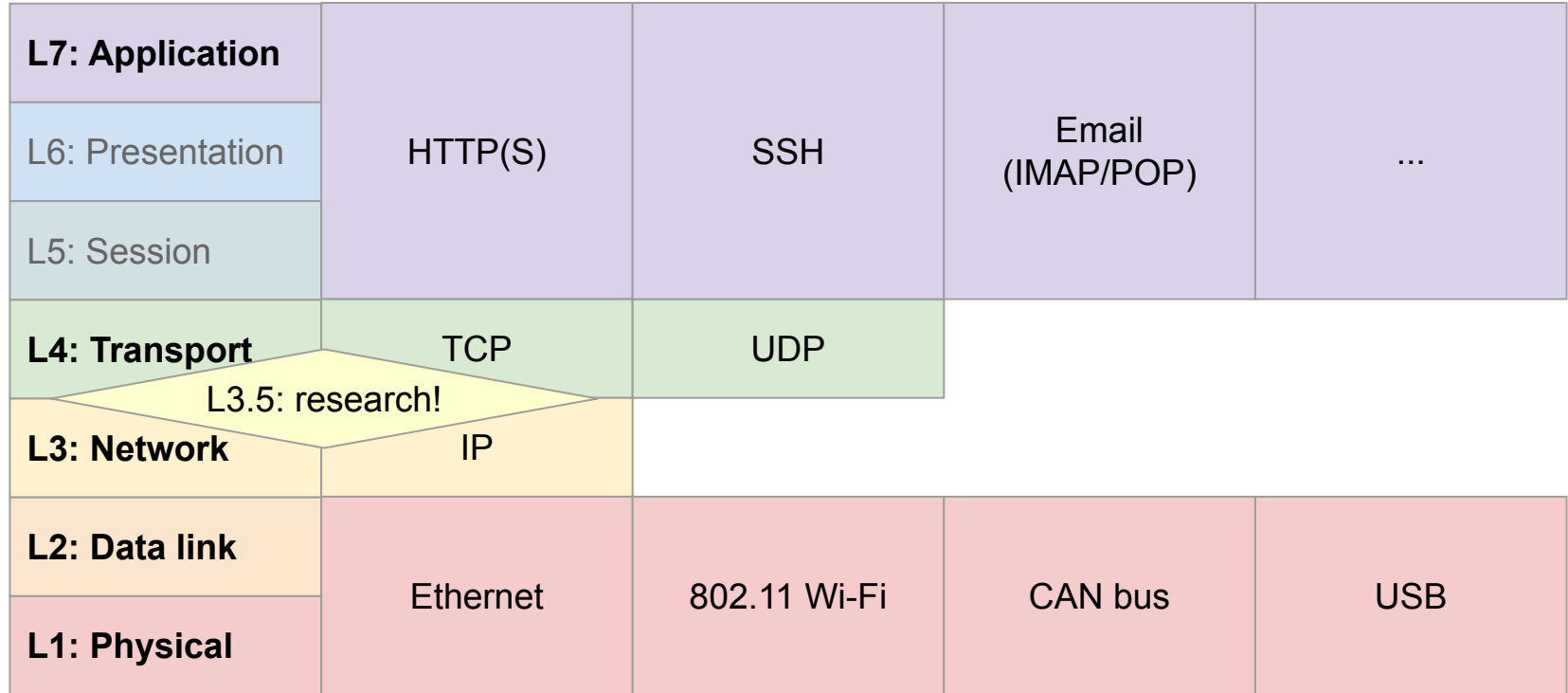
Agenda: Layering and Sockets!

- Layering
- Sockets
 - Establishing connections
 - Send and receive

Layering

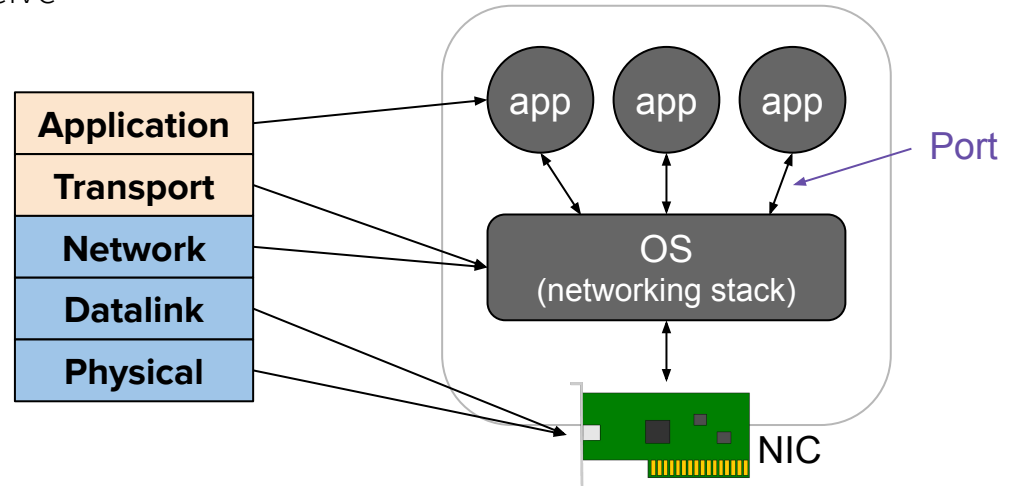
L7: Application	do the thing
L6: Presentation	(ignored here)
L5: Session	(ignored here)
L4: Transport	beyond delivery: (un)reliability, packet assembly, congestion control, ...
L3: Network	global delivery, best-effort
L2: Data link	local delivery, best-effort
L1: Physical	physical transfer of bits

Layering in practice



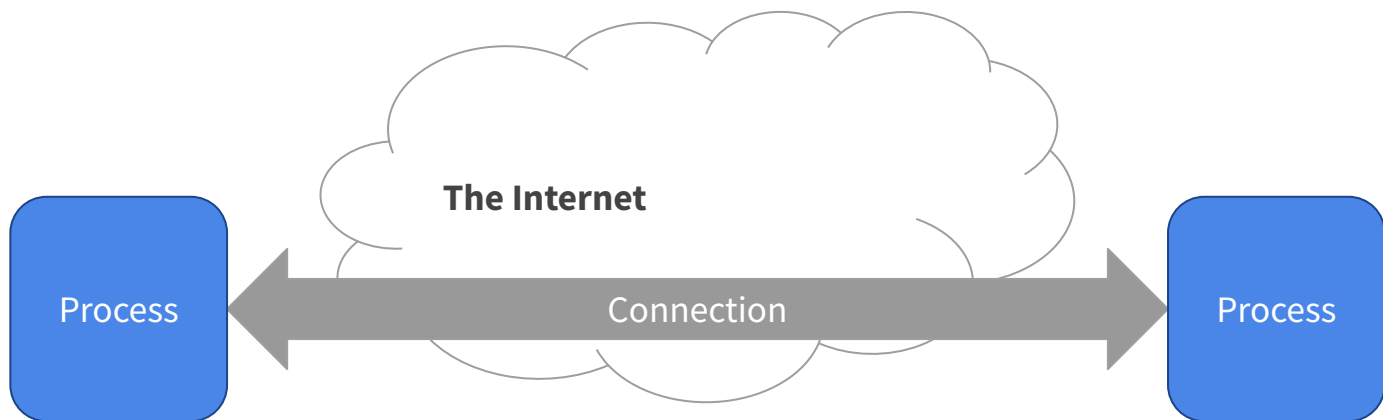
Sockets

- Developed here, at UC Berkeley!
- OS abstraction for **connections**
- Allow L7 applications to operate on data streams (not packets)
 - Connect, listen, accept, send, receive
- Open a socket between:
 - Source IP address : *port*
 - Destination IP address : *port*



Connection (the basic abstraction)

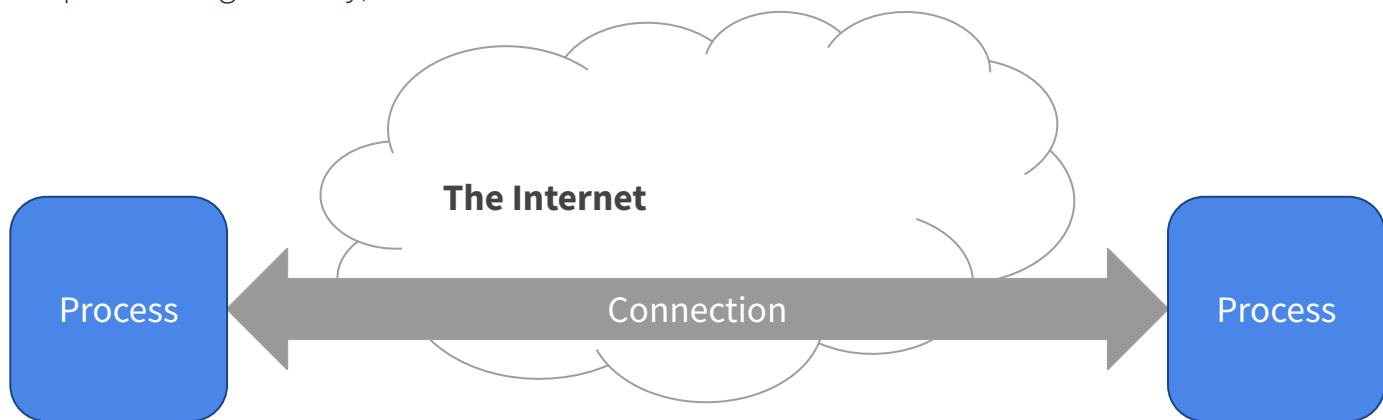
- Pipes data between two processes (on different hosts)
- Data flows both ways!



Connection (the basic abstraction)

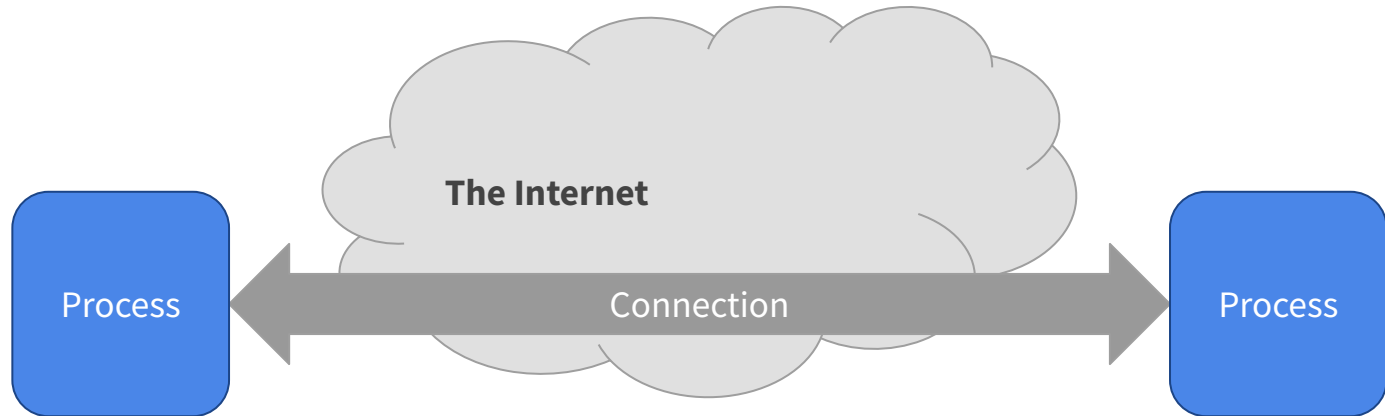
- Data is sent simply as a stream of bits
- Reconstruction of bits only at the endpoints
- The Internet knows nothing* about what it's transmitting!

* (unless you're implementing security)



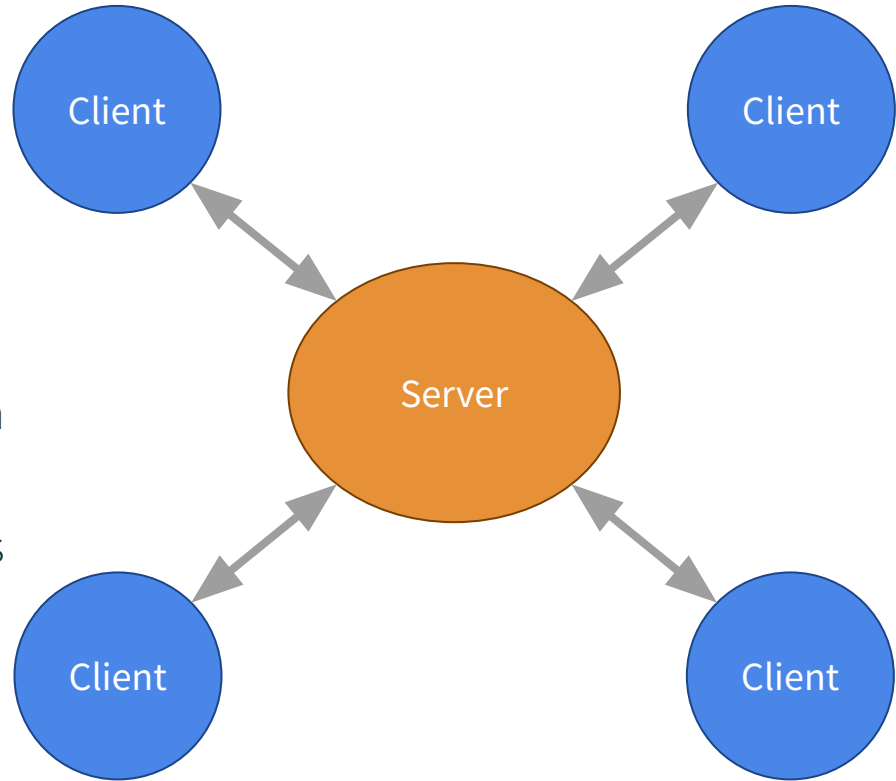
Socket API

- Establish Connection
- Sending
- Receiving



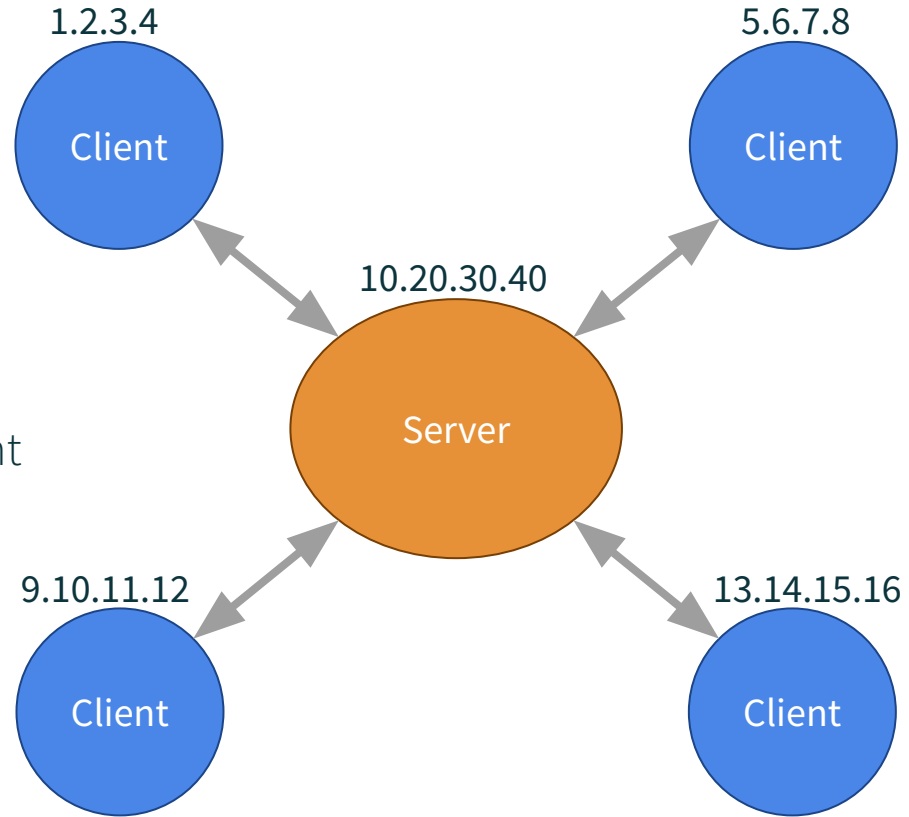
Connections

- Two types of sockets
 - Server and Client
- Servers *listen* for clients to connect to them
 - Wait until a connection is attempted
 - Accept and dispatch connection
 - Usually serving many clients at once
- Clients *initiate* new connections to servers
- Example
 - Server: berkeley.edu
 - Client: Your internet browser



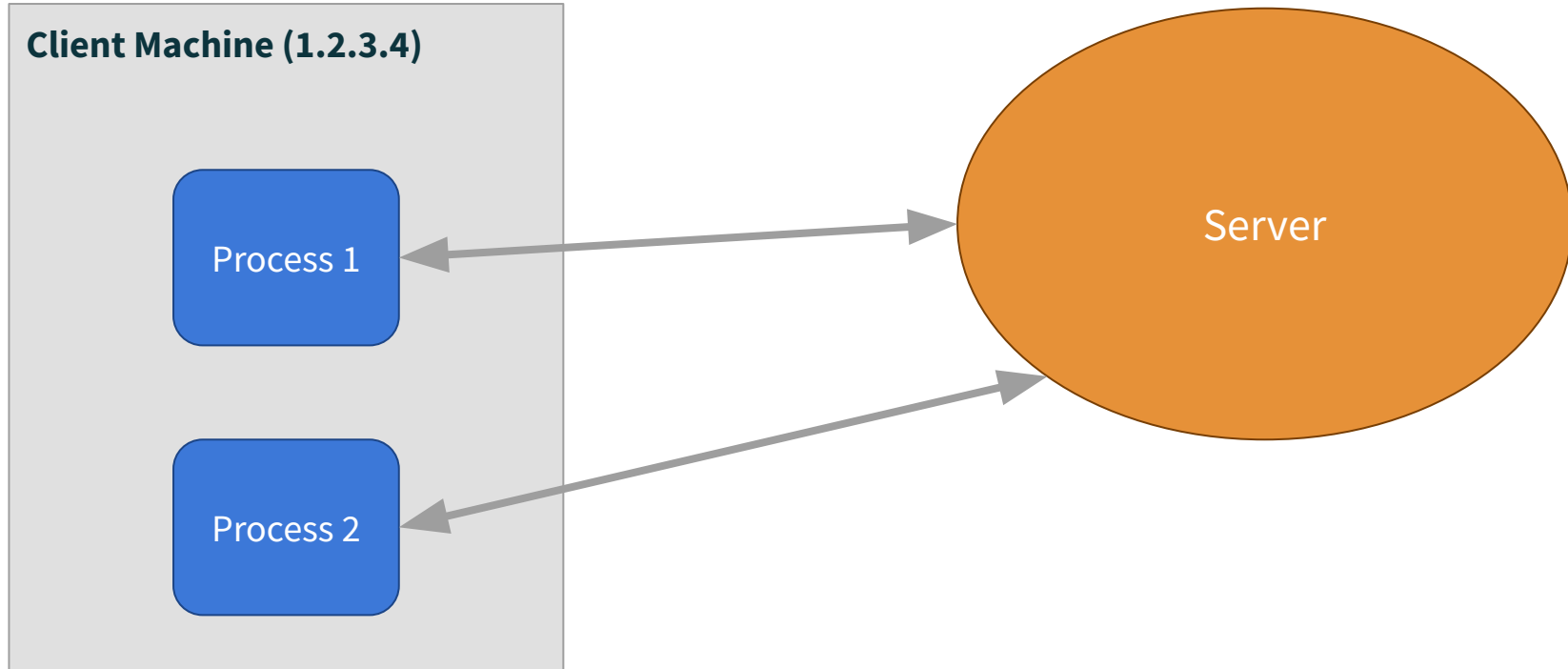
Connections

- Hosts have addresses
 - Unique identifier (just like a street address)
- Clients (different users) find servers with their addresses
 - Servers send data back with the client address
- Example addresses →
Are addresses enough to make this work?



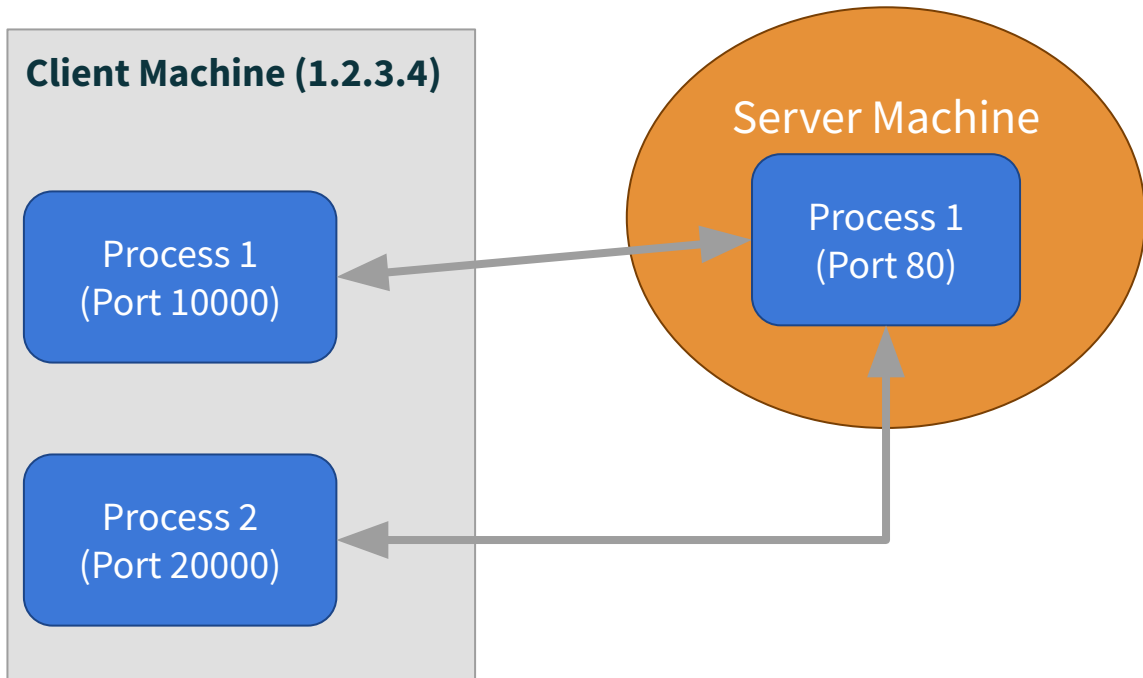
Address aren't enough

How does the client computer know which process (i.e. web browser) to deliver data to?



Ports

- Sockets are identified by unique IP:port pairs
- A port is a number that the OS associates with a socket when it is created
 - i.e. sending to address “1.2.3.4:10000” would send data to the socket owned by Process 1



Ports

- Packets carry port number
- Servers listen on a port
 - Which one depends on application
 - HTTP: 80
 - SSH: 22
- Client process connects to well known port
- Client also has a port
 - Randomly assigned by OS
 - Used by OS to send data to correct process

