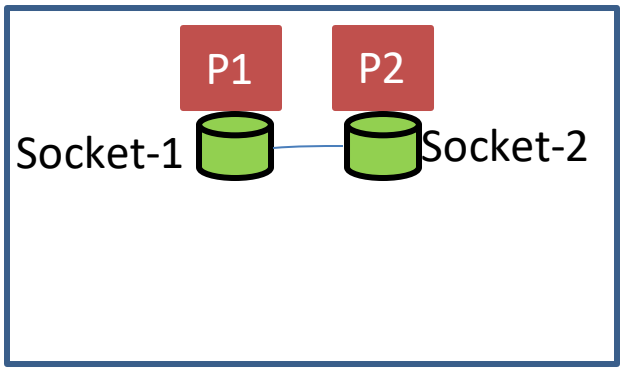
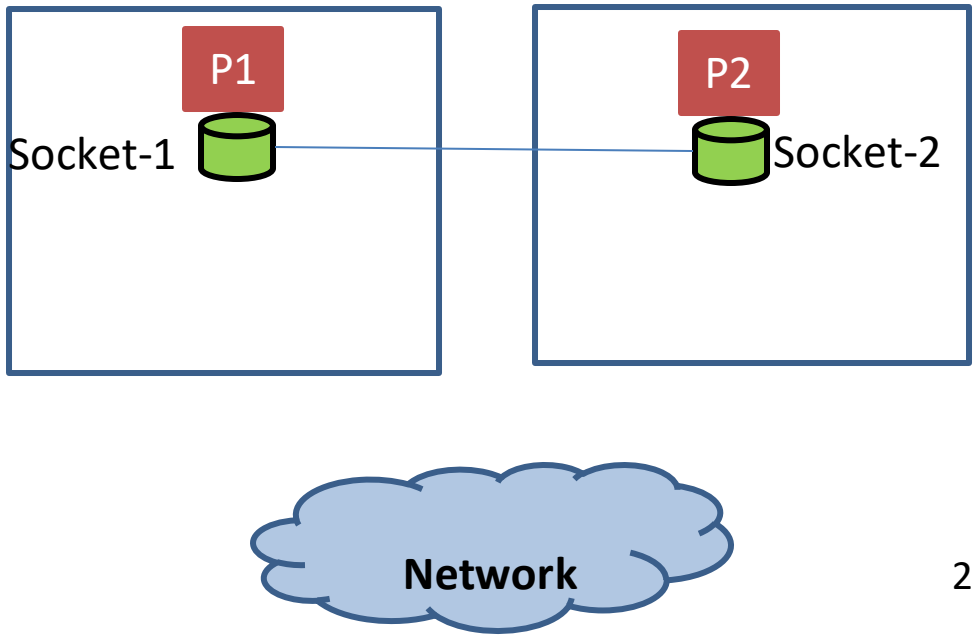
Introduction to Socket Programming

CS 4450

What is a Socket?

- A socket is a method for accomplishing inter-process communication (IPC)
  - Allows one process to communicate with another process on the same or different machine

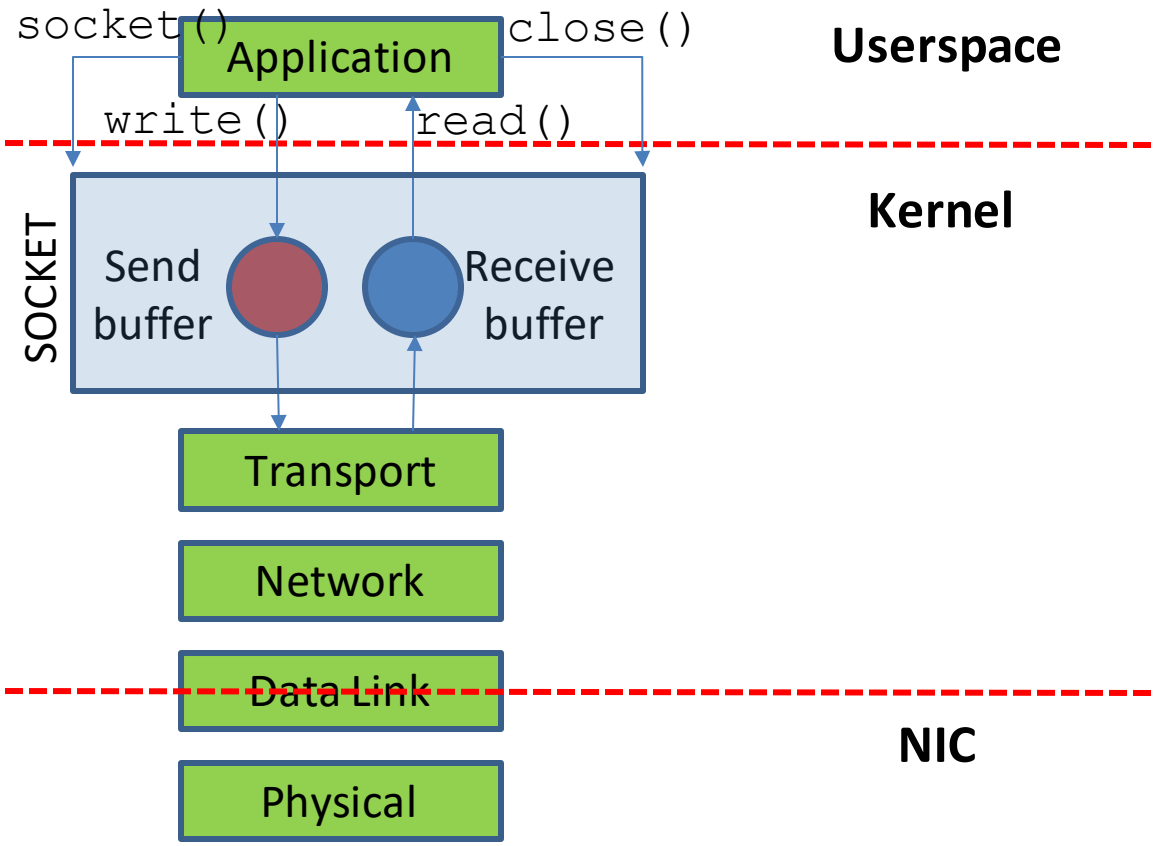




Operations on a Socket

- Socket works very similar to a file
  - `open()` `socket()` -- open a socket
  - `read()` -- read from a socket (analogous to receive data)
  - `write()` -- write to a socket (analogous to send data)
  - `close()` -- close the socket

Where does Socket fit in the Network Stack?



Blocking and Non-blocking Sockets

- By default `read()` and `write()` operations are blocking
  - Function does not return until the operation is complete

- `read()` blocks until there is some data available in the receive buffer

- When does `write()` block?
  - When the send buffer is full

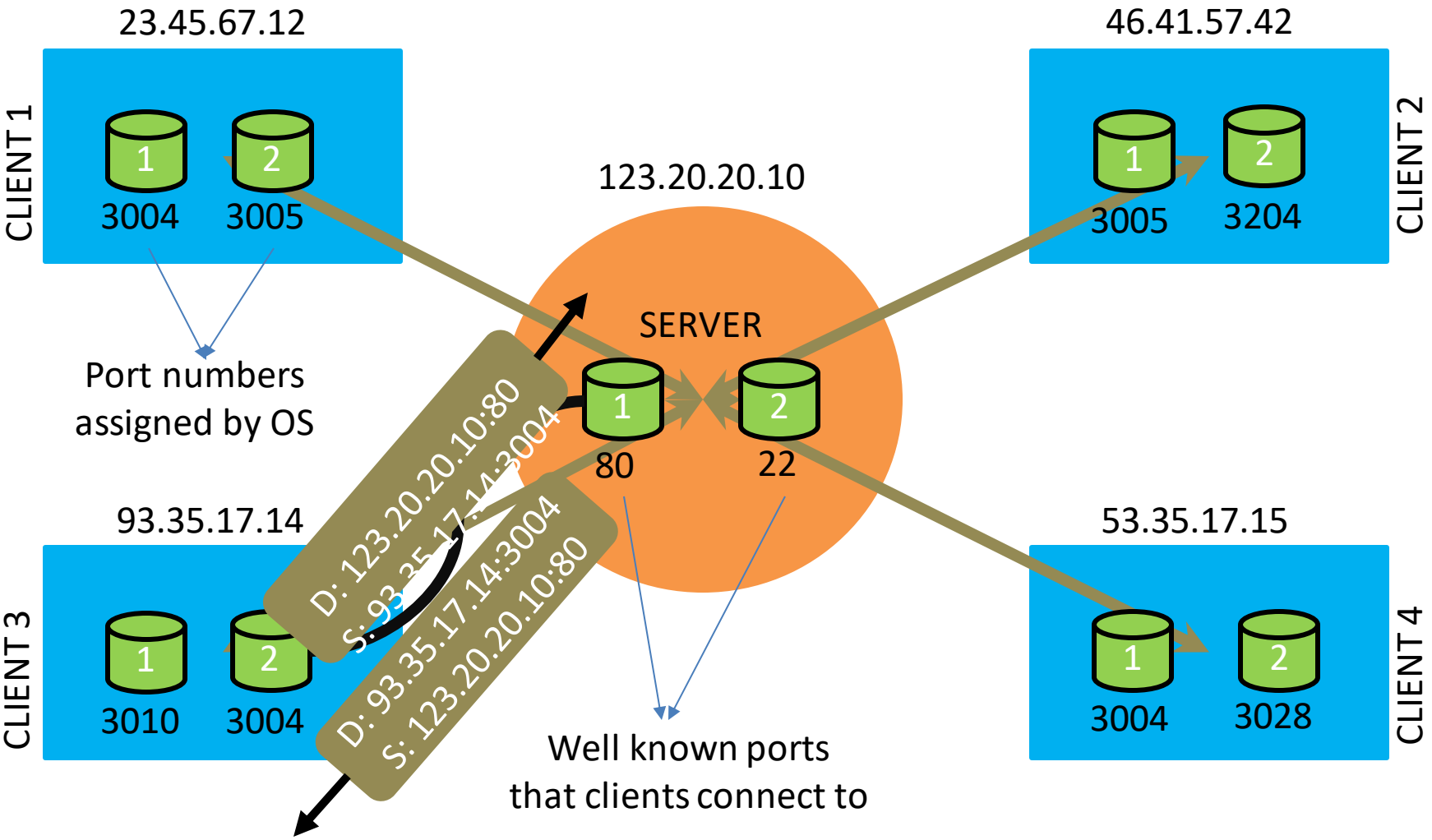
Blocking and Non-blocking Sockets

- Non-blocking `read()` and `write()` return immediately

- `read()`
  - If there is some data in receive buffer, `read()` succeeds and returns the amount of data read
  - If the receive buffer is empty, `read()` returns the ERROR code

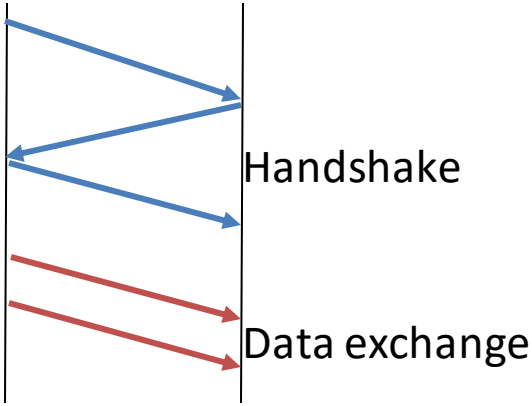
- `write()`
  - If there is some space available in the send buffer, `write()` succeeds and returns the amount of data written
  - If the send buffer is full, `write()` returns the ERROR code

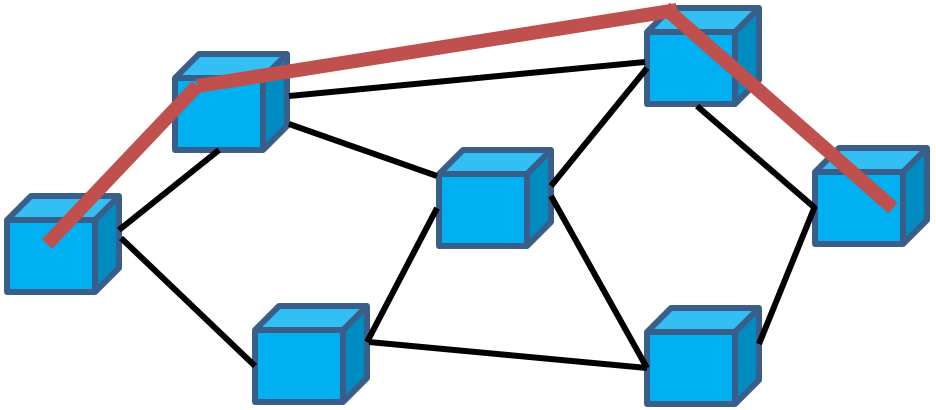
Client-Server Model



Two traditional modes of communication

- Connection-oriented Communication
  - Establish a logical or physical connection before exchanging data

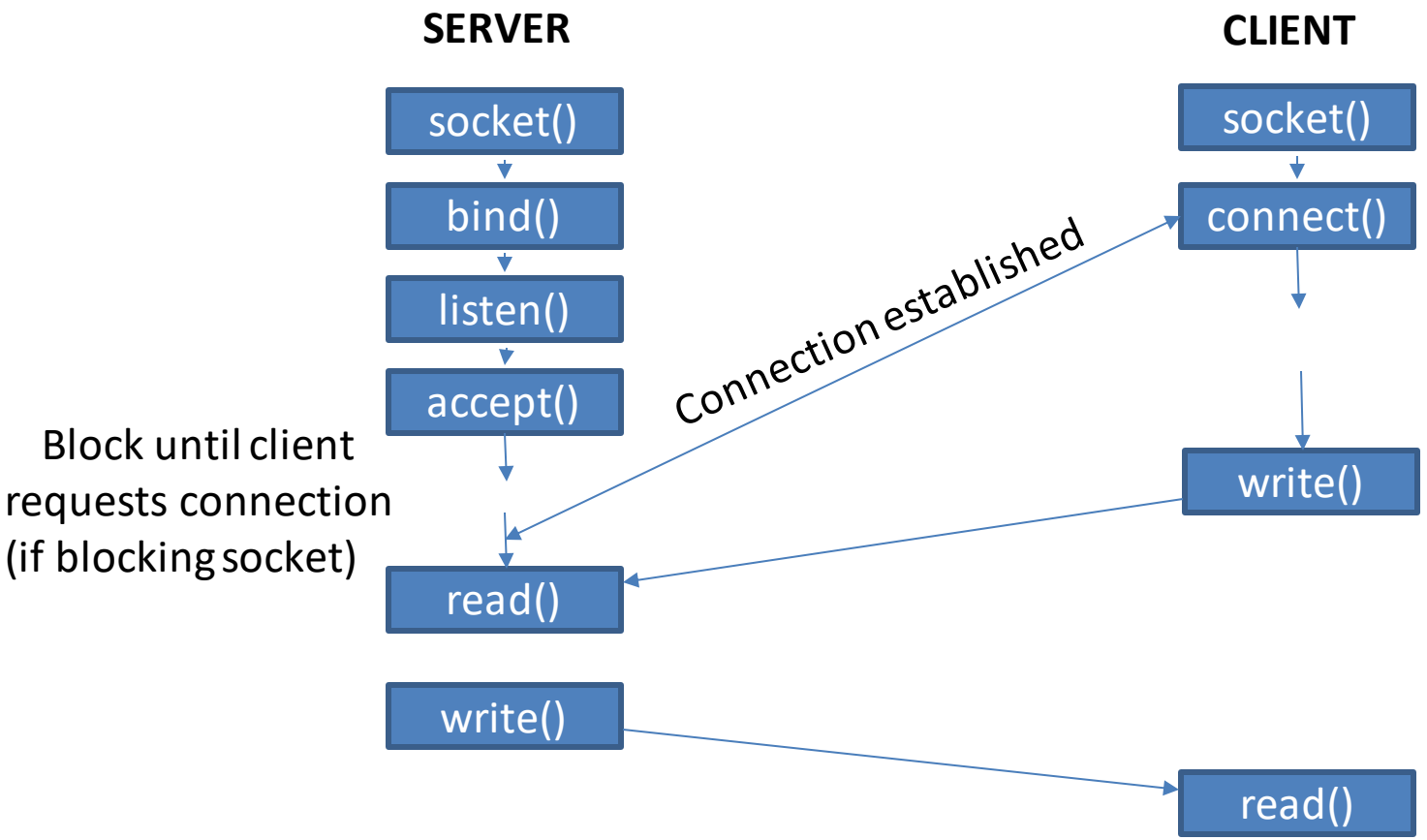




- Connectionless Communication
  - Start exchanging data without any prior arrangements between endpoints

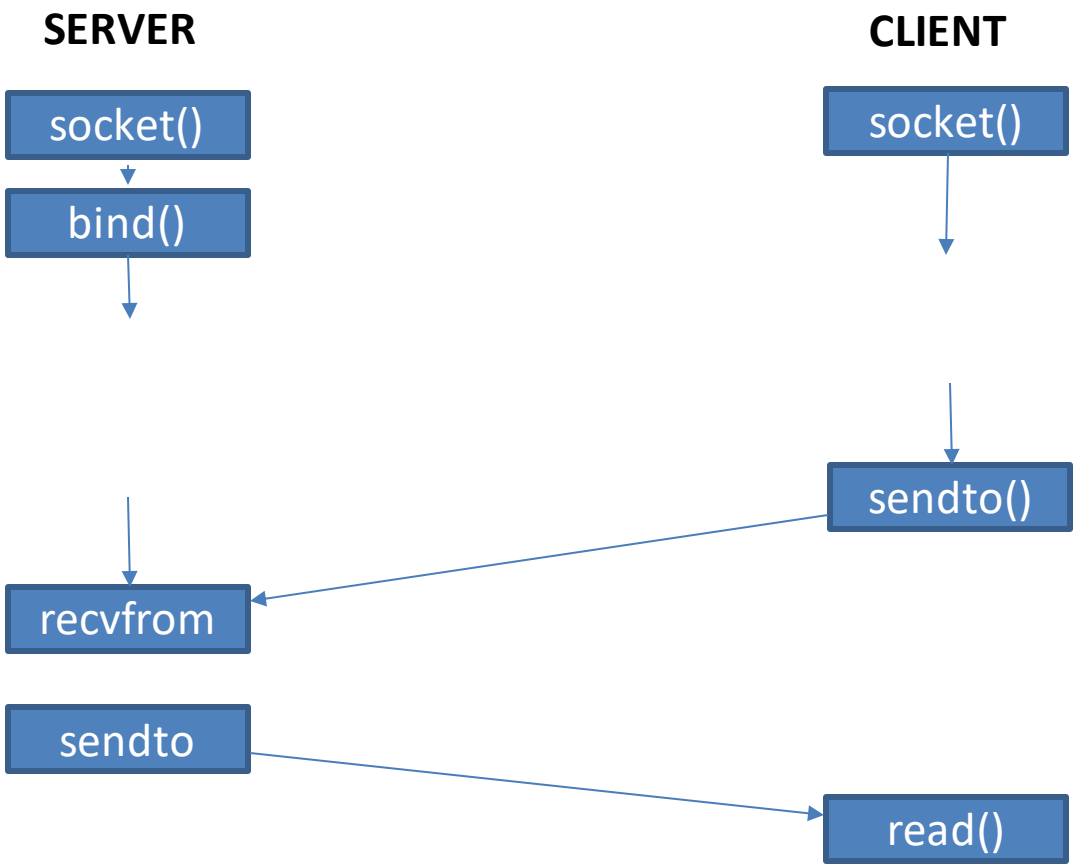
Client-Server Model - APIs

- Connection-oriented protocol (TCP-suite)



Client-Server Model - APIs

- Connectionless protocol (UDP-suite)



Questions?

Demos

Thank you!