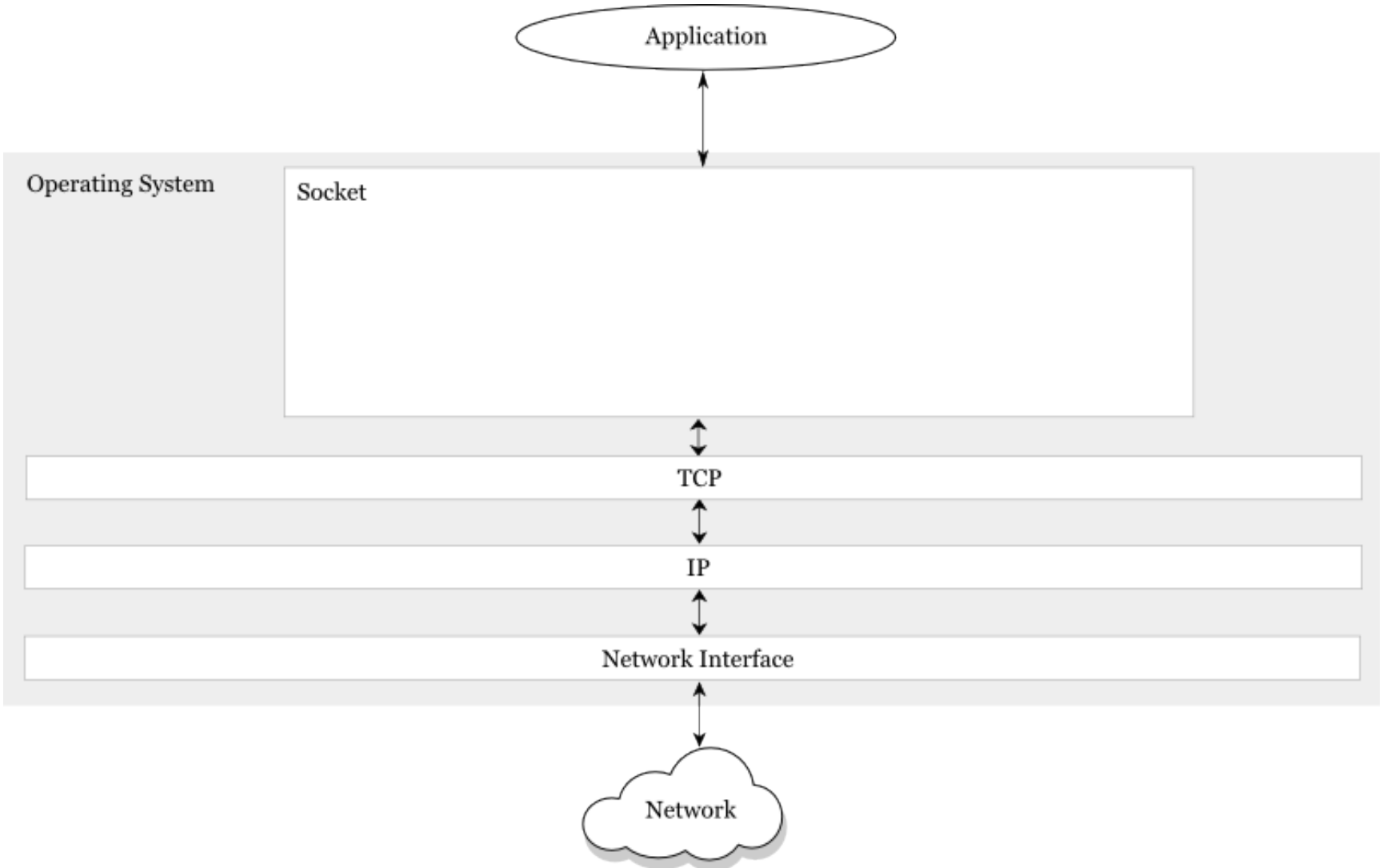
Sockets and the OS

Daniel Zappala

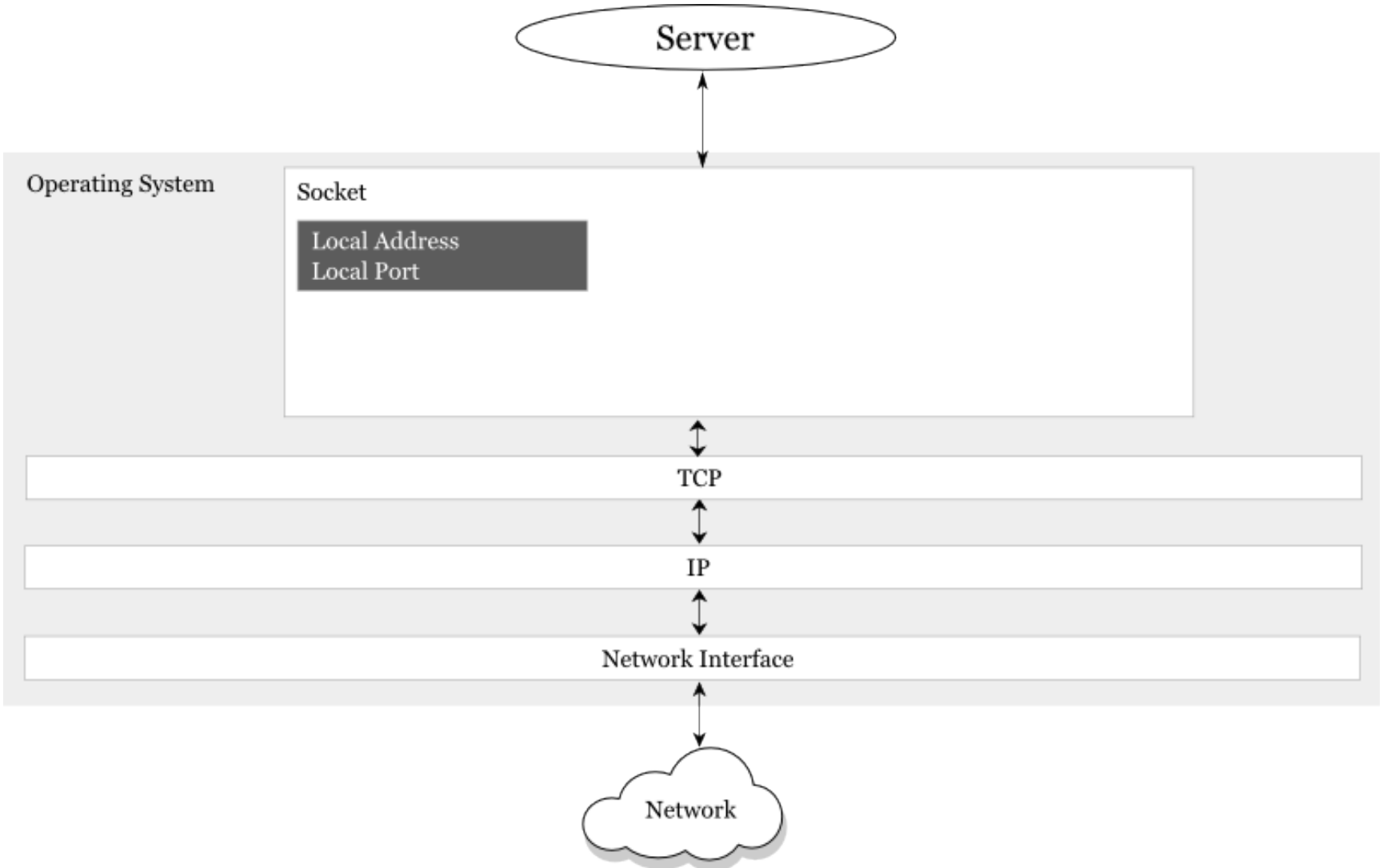
CS 360 Internet Programming
Brigham Young University

socket(): create interface to the transport protocol



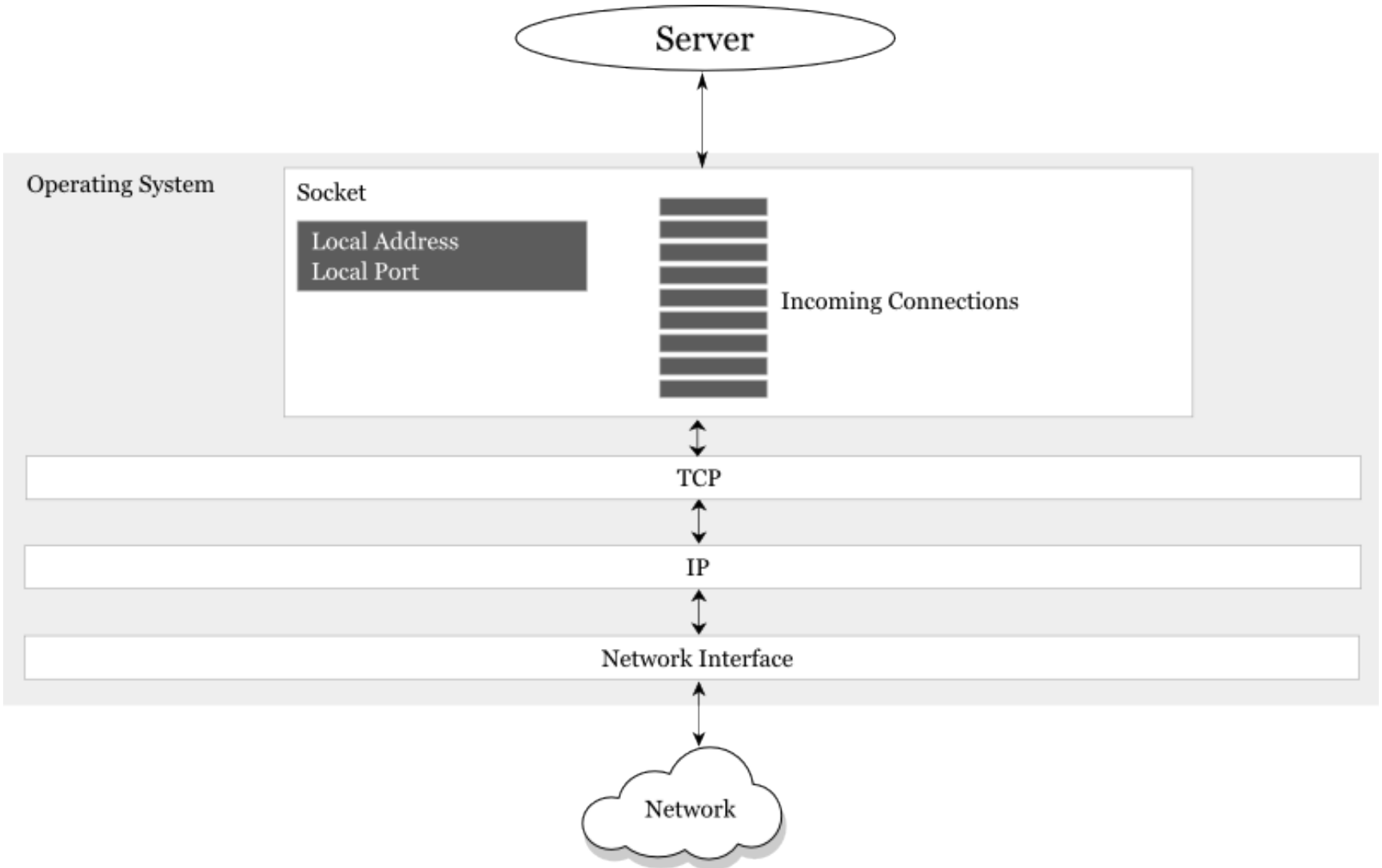
- initialize data structures

bind(): establish local address and port



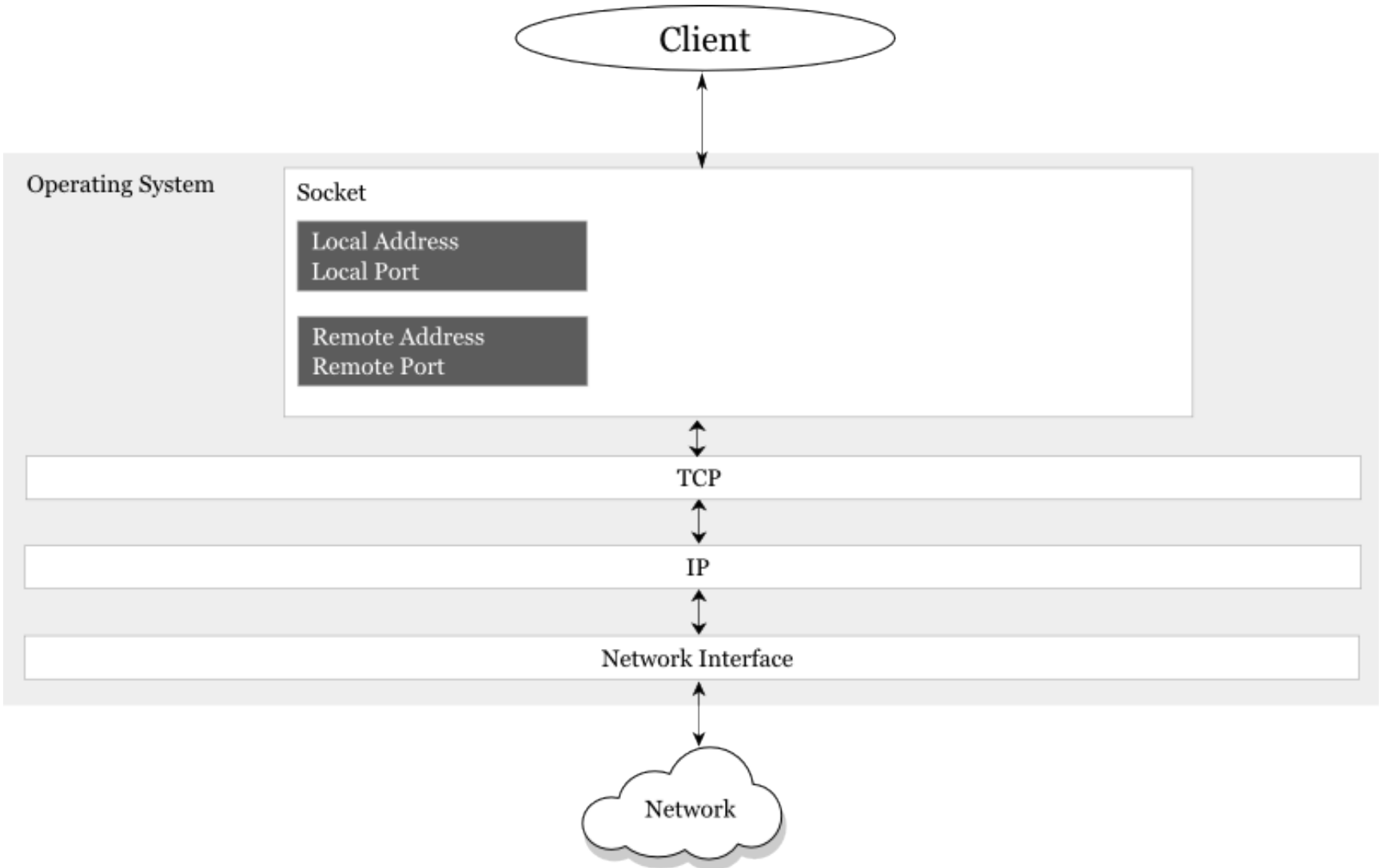
- address/port combination must be unique

listen(): convert to a listening socket



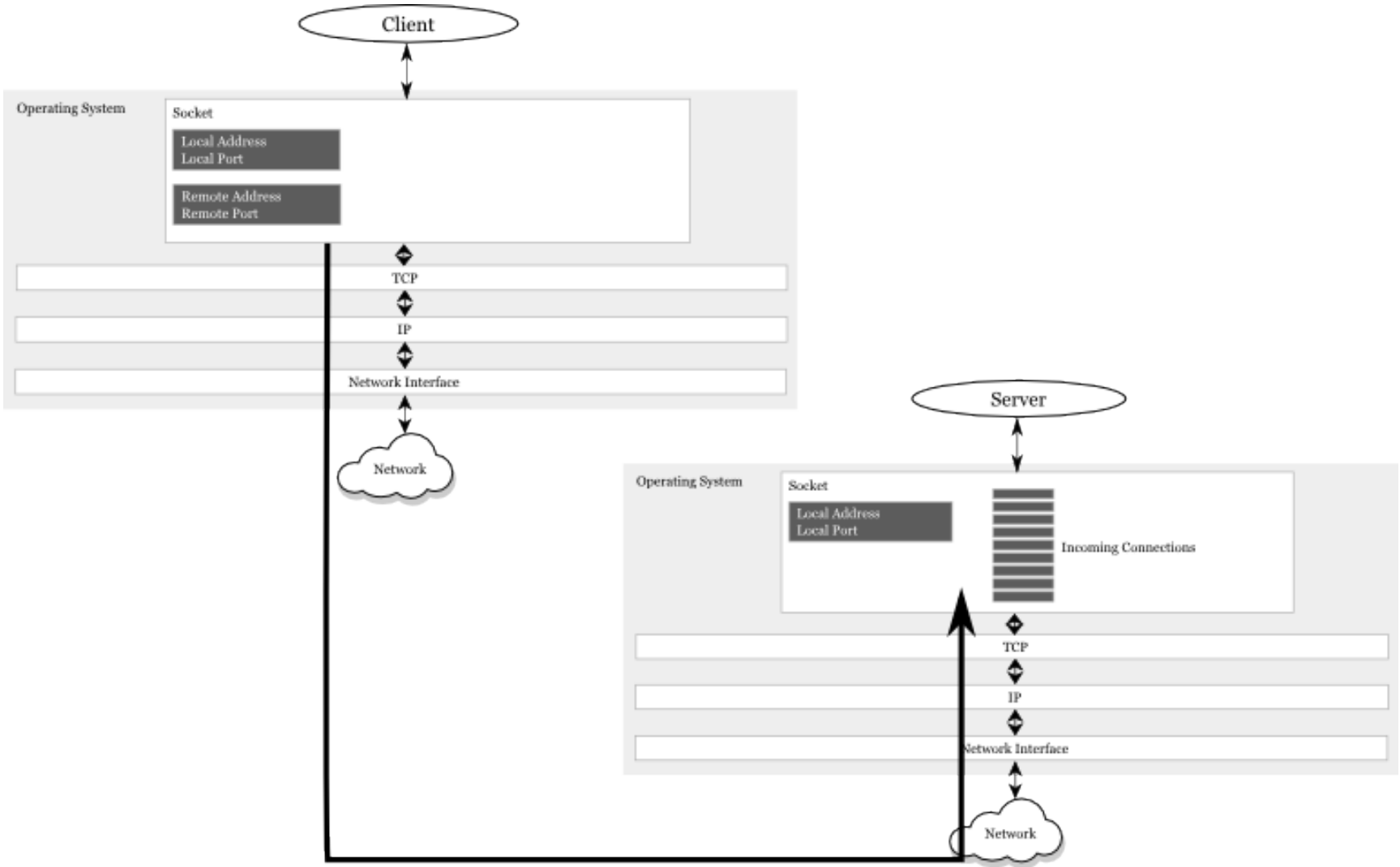
- creates queue for incoming connections

connect(): bind addresses and ports



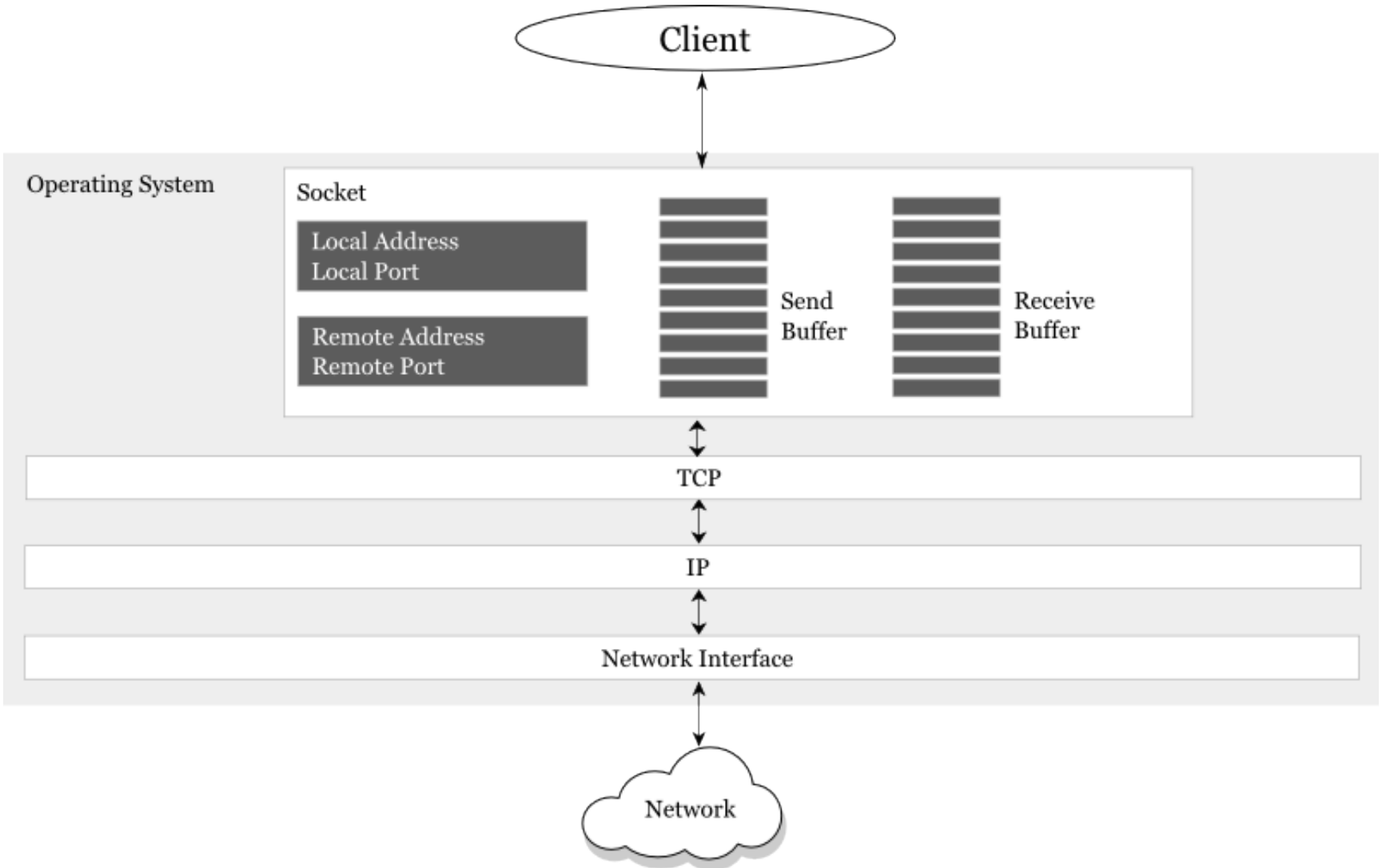
- four-tuple must be unique for a connected socket

connect(): connect to remote server



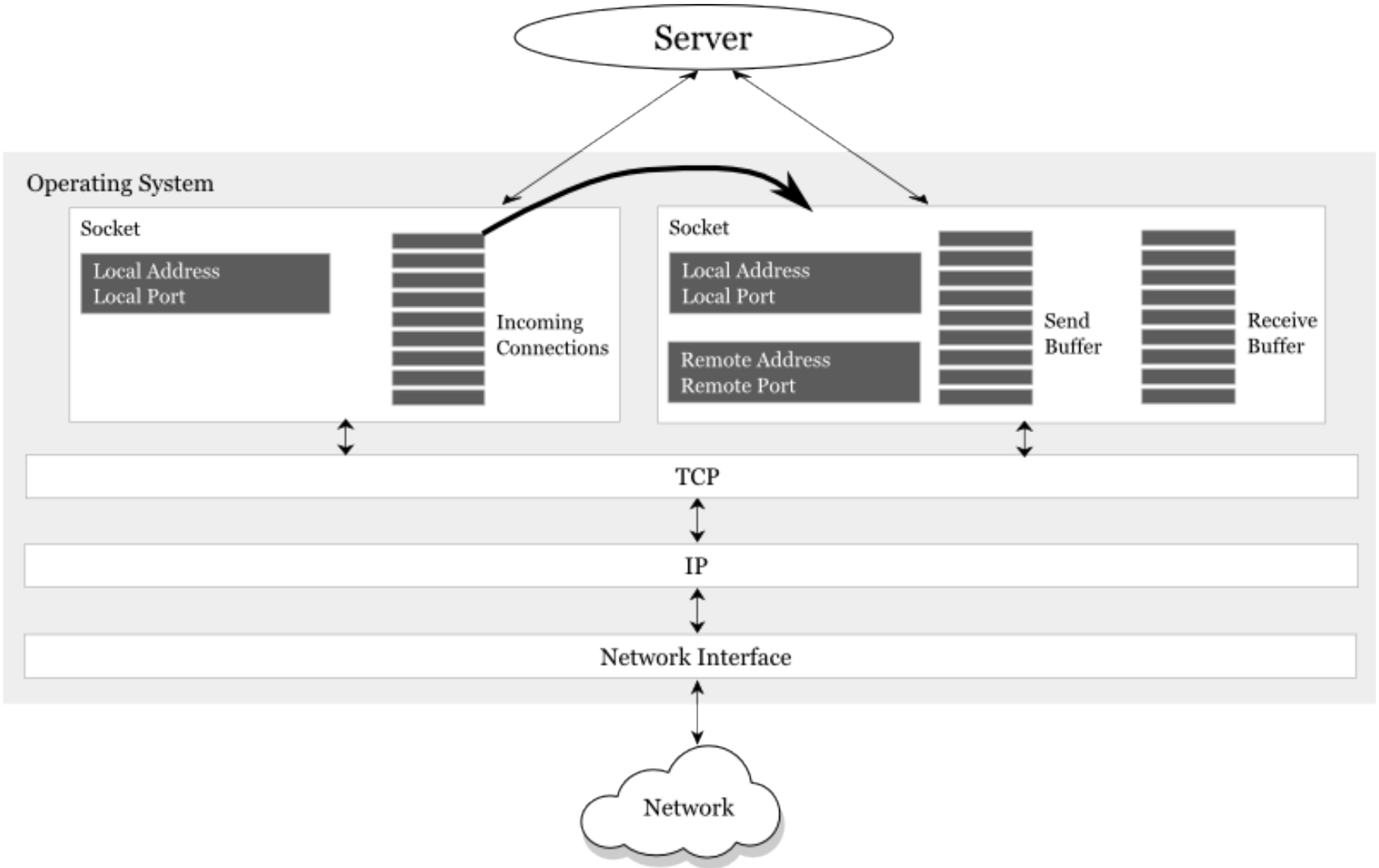
- TCP establishes a connection

connect(): create send and receive buffers



- allocates buffers when connection is completed

accept(): create a new socket from incoming connections

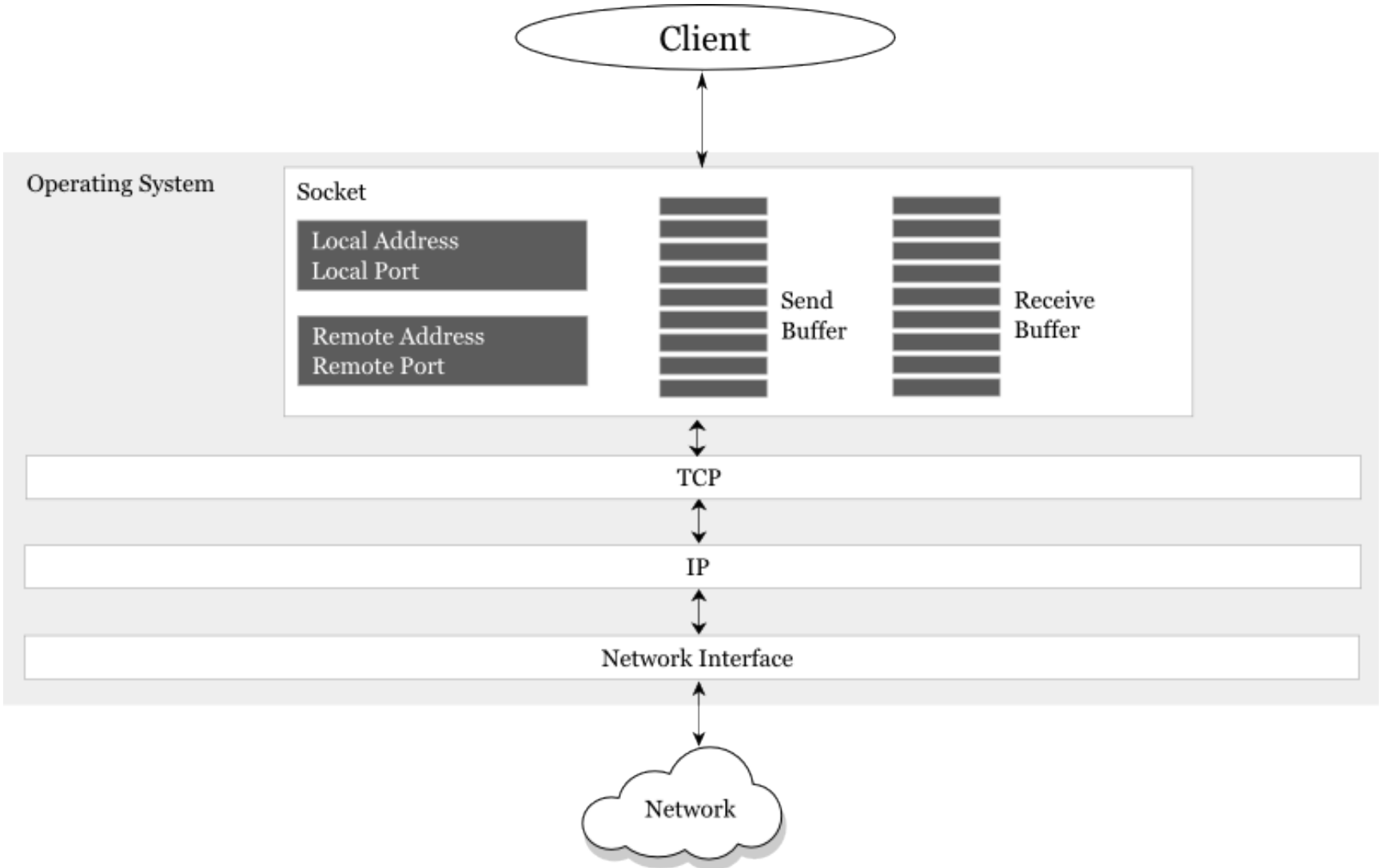


- allocates a new socket with buffers, four-tuple is unique

accept(): sockets

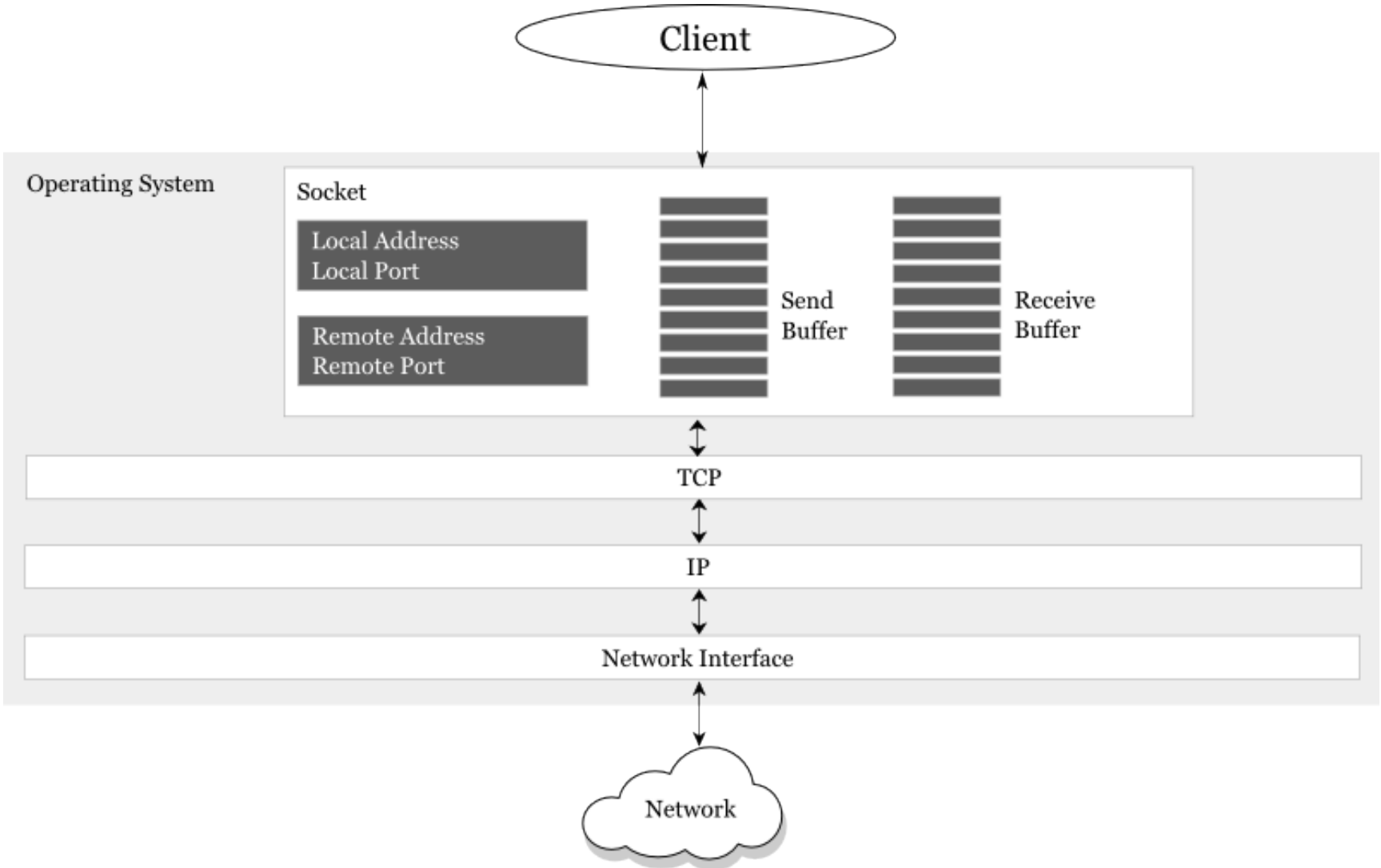
- listening socket is bound to just a local address and port
  - no other socket may listen with this combination
  - this socket *only* accepts incoming connections
- each connected socket is bound to local address and port *and* remote address and port
  - each connected socket must have a unique four-tuple
  - they all use the same local port as the listening socket
  - operating system demultiplexes incoming packets and directs them to the appropriate socket
- so a web server has
  - one socket listening for incoming connections on port 80
  - one socket for each connected client, each also on port 80, but with a different remote address/port combination
  - juggling these sockets (with threads or asynchronous I/O is the major job of all servers

send(): put data into send buffer



- returns when data stored, not when data delivered
- TCP in charge of sending data, regulating rate

recv(): get data from receive buffer



- blocks until data available, by default
- we will use non-blocking I/O later