#In the name of Allah

Computer Engineering Department
Sharif University of Technology

CE443- Computer Networks

Socket Programming

Acknowledgments: Lecture slides are from Computer networks course thought by Jennifer Rexford at Princeton University. When slides are obtained from other sources, a reference will be noted on the bottom of that slide.

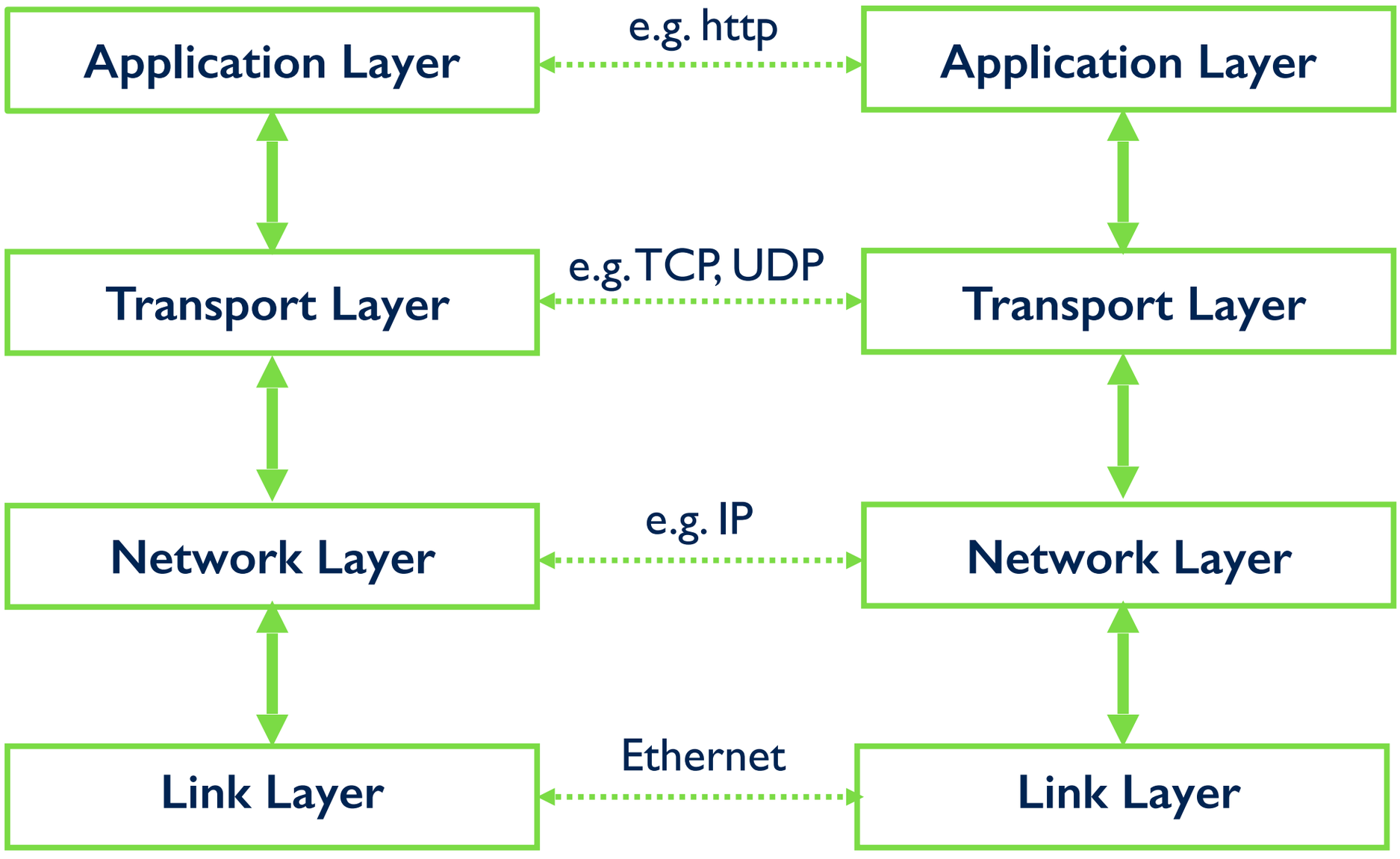
#What is Socket?

- Networking in UNIX is I/O
- A way to speak to other programs using standard Unix file descriptors, here a file is a Network Connection.
- Socket Programming is a way to create and Handle a file for a network connection.
- Sockets can be different …
  - DARPA Internet addresses, Internet Sockets.

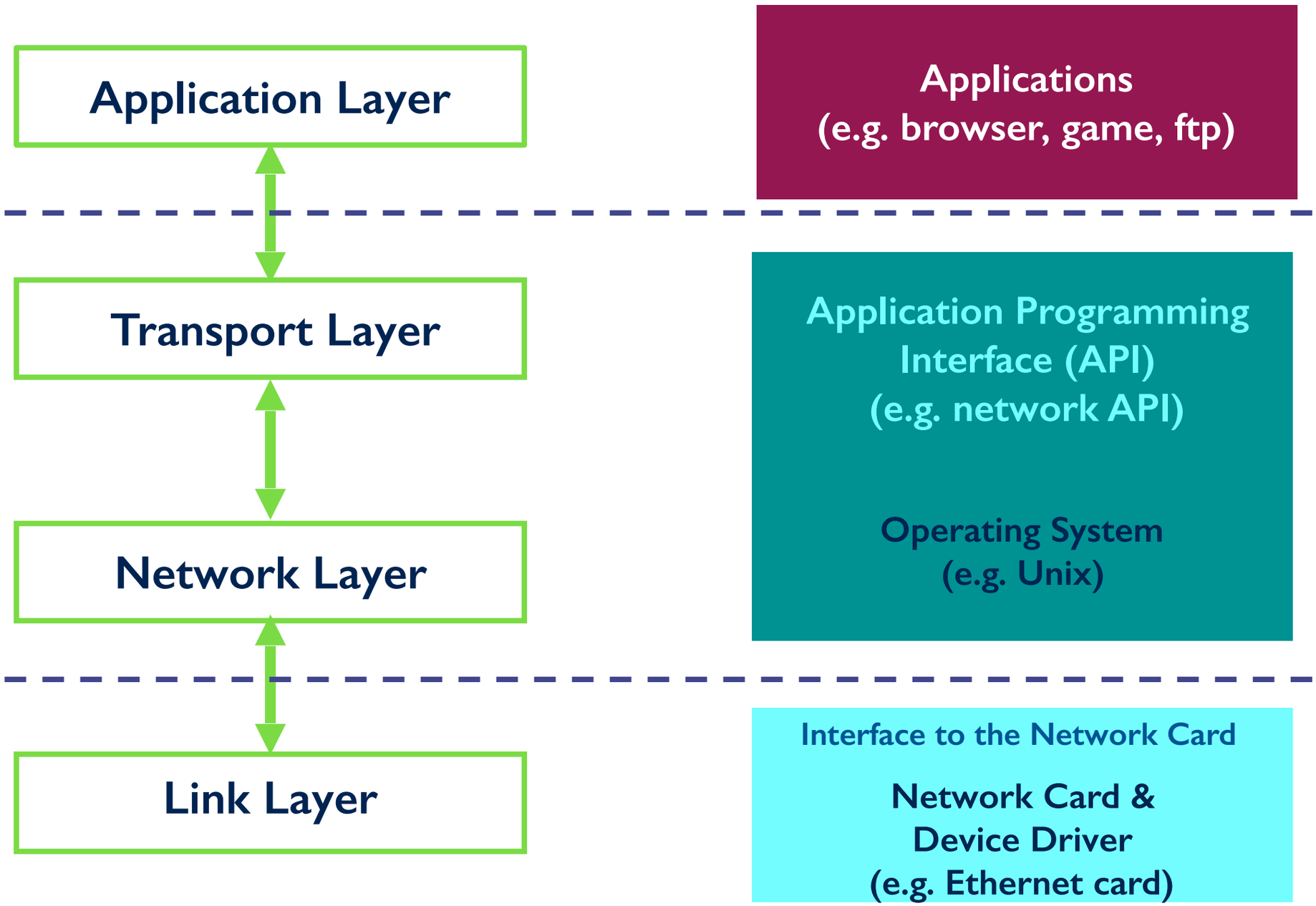
#Internet Sockets

- Support stream and datagram packets (e.g. TCP, UDP, IP)
- Is Similar to UNIX file I/O API (provides a file descriptor)
- Based on C, single thread model (does not require multiple threads)

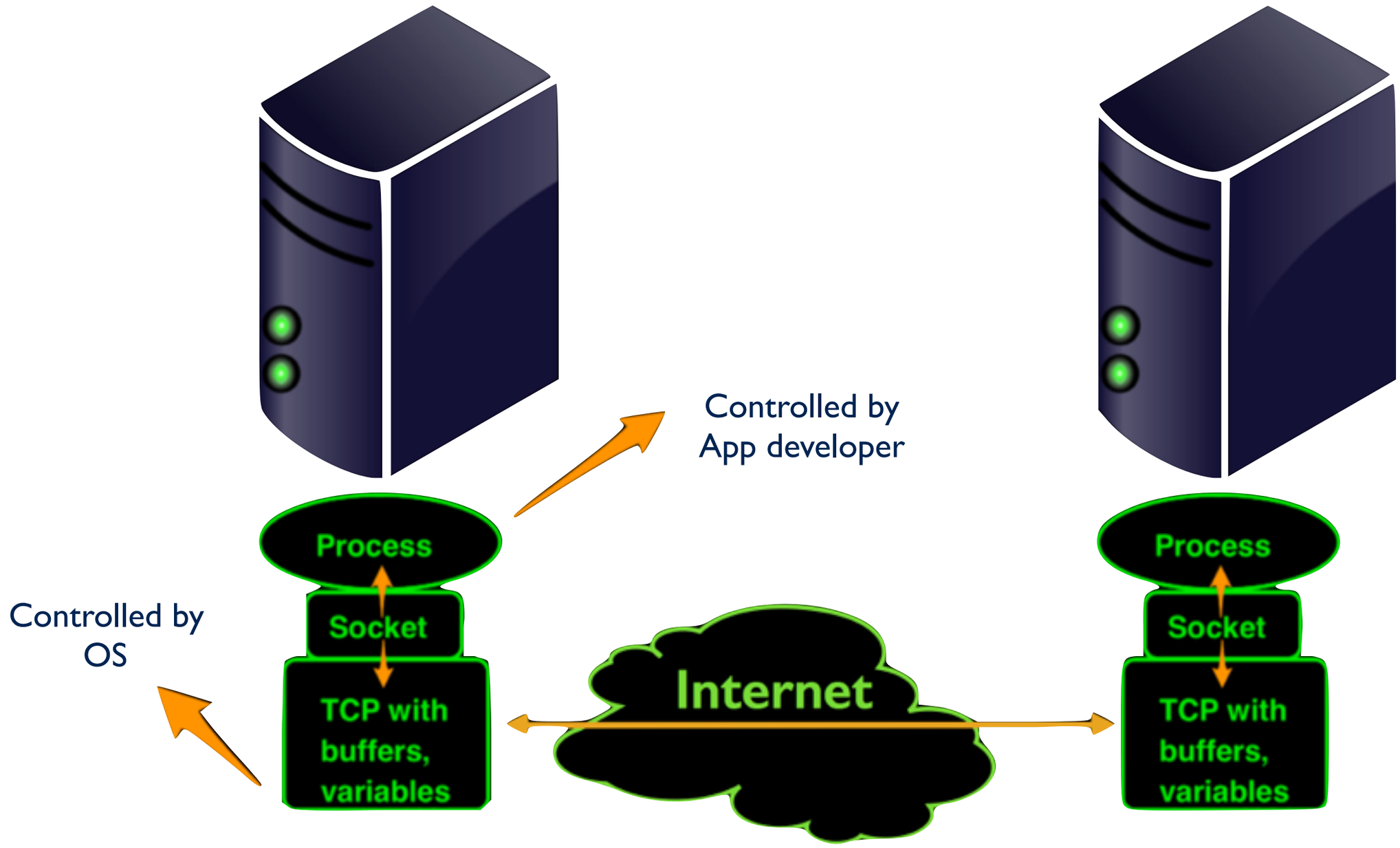
#Layers of the IP Protocol Suite



#Protocol Suite Location



#Socket Programming API



#Delivering the Data: Division of Labor

Application

- Read data from and write data to the socket
- Interpret the data (e.g., render a Web page)

Operating System

- Deliver data to the destination socket
- Based on the destination port number

Network

- Deliver data packet to the destination host
- Based on the destination IP address

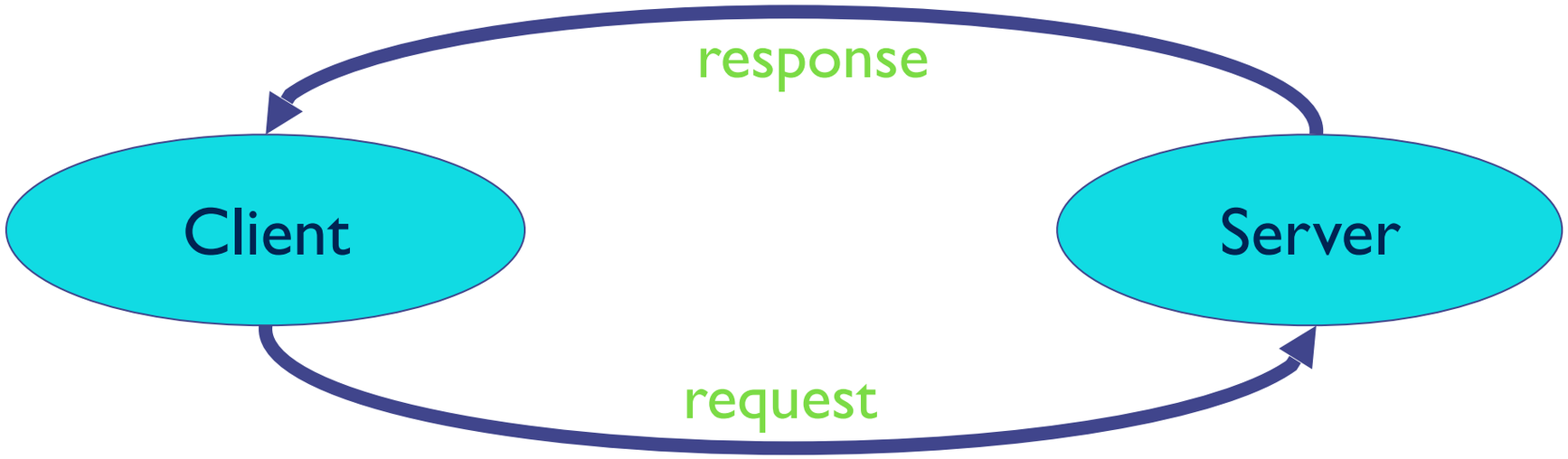
#Identifying the Receiving Process

- Sending process must identify the receiver
  - The receiving end host machine
  - The specific socket in a process on that machine
- Receiving host
  - Destination address that uniquely identifies the host
  - An IPv4 address is a 32-bit quantity
- Receiving socket
  - Host may be running many different processes
  - Destination port that uniquely identifies the socket

#Socket Programming: Naming and Addressing

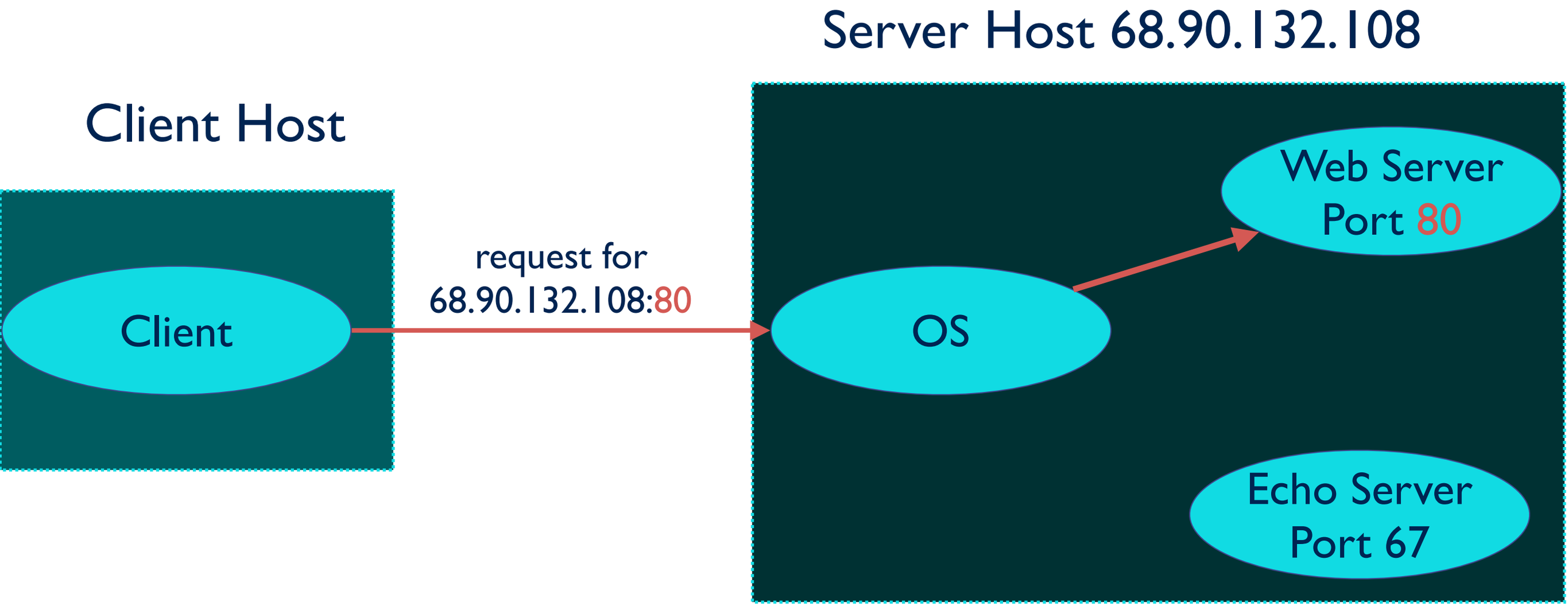
- Host name
  - identifies a single host (Recall: DNS)
  - variable length string
  - is mapped to one or more IP addresses
- IP Address
  - written as dotted octets (e.g. 10.0.0.1)
  - 32 bits. Not a number! But often needs to be converted to a 32-bit number to use.
- Port number
  - identifies a process on a host
  - 16 bit number

#Client-Server Architecture

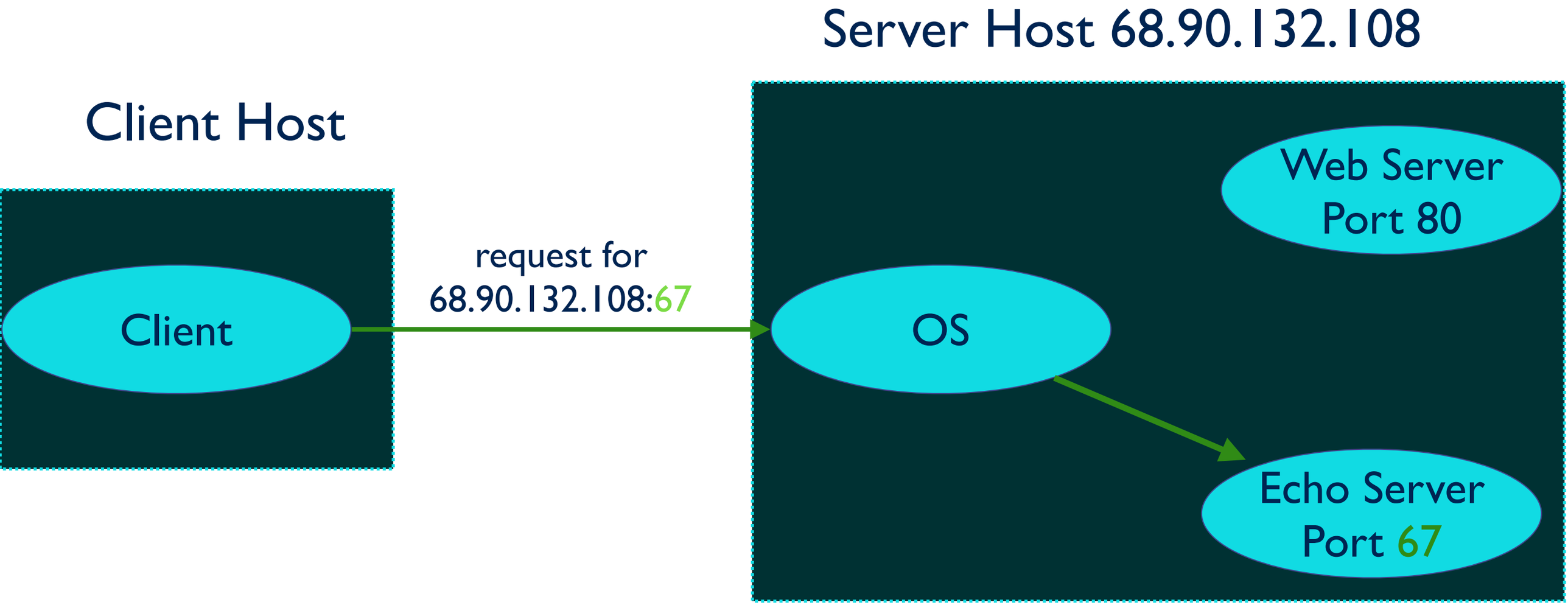


- Client requests service from server
- Server responds with sending service or error message to client
- What if Server starts a connection?

#Identifying the Receiving Process



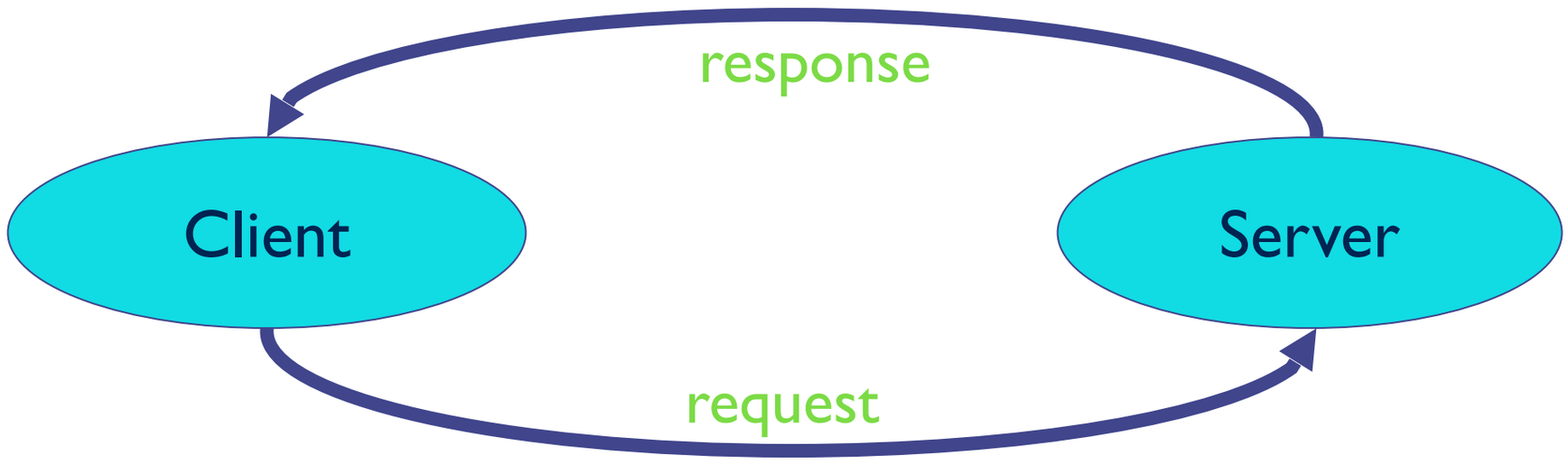
#Identifying the Receiving Process



Port Numbers are Unique on Each Host…

- Port number uniquely identifies the socket:
  - Cannot use same port number twice with same address
  - Otherwise, the OS can't demultiplex packets correctly
- Operating system enforces uniqueness
  - OS keeps track of which port numbers are in use
  - Doesn't let the second program use the port number

#nc!



- **nc** : arbitrary TCP and UDP connections and listens!
- now we just focus on a very very simple example:
- nc hostname port
- nc -l port

UNIX Socket API

#UNIX Socket API

- Socket interface
  - Originally provided in Berkeley UNIX
  - Later adopted by all popular operating systems
  - Simplifies porting applications to different OSes(even to the Windows!)
- In UNIX, everything is like a file
  - All input is like reading a file
  - All output is like writing a file
  - File is represented by an integer file descriptor
- API implemented as system calls
  - E.g., connect, read, write, close, …

#Two Types of Internet Sockets

Connection-oriented sockets

- How to snd/rcv data
- How to establish connection
  - 3-way handshake?
- How to identify socket
- How to create socket
- How to close socket

Type of socket: stream socket

Connectionless sockets

- How to snd/rcv data
- How to identify socket
- How to create socket
- How to close socket

Type of socket: datagram socket