#In the name of Allah

Computer Engineering Department
Sharif University of Technology

CE443- Computer Networks

Socket Programming
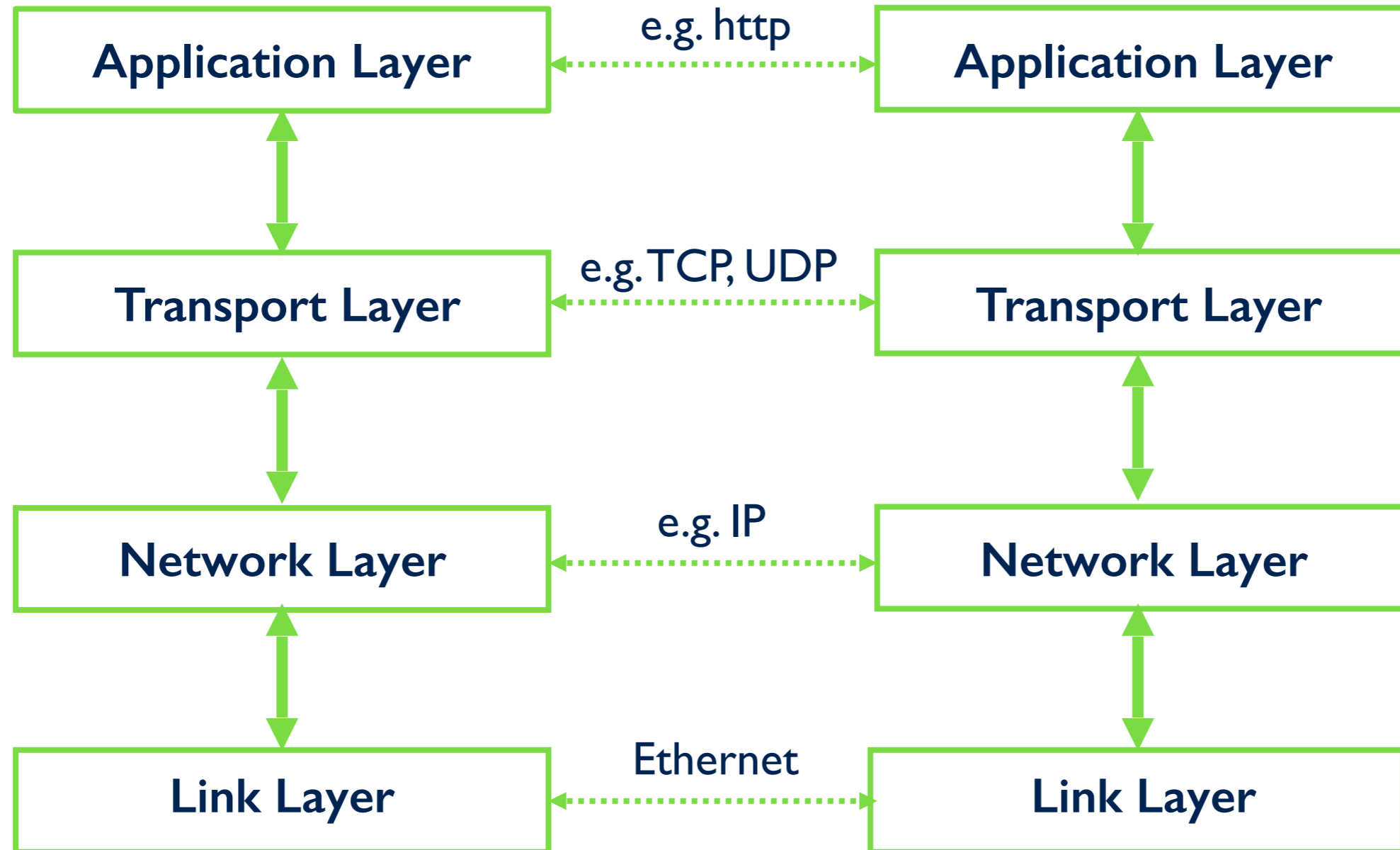
#What is Socket?

- 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)

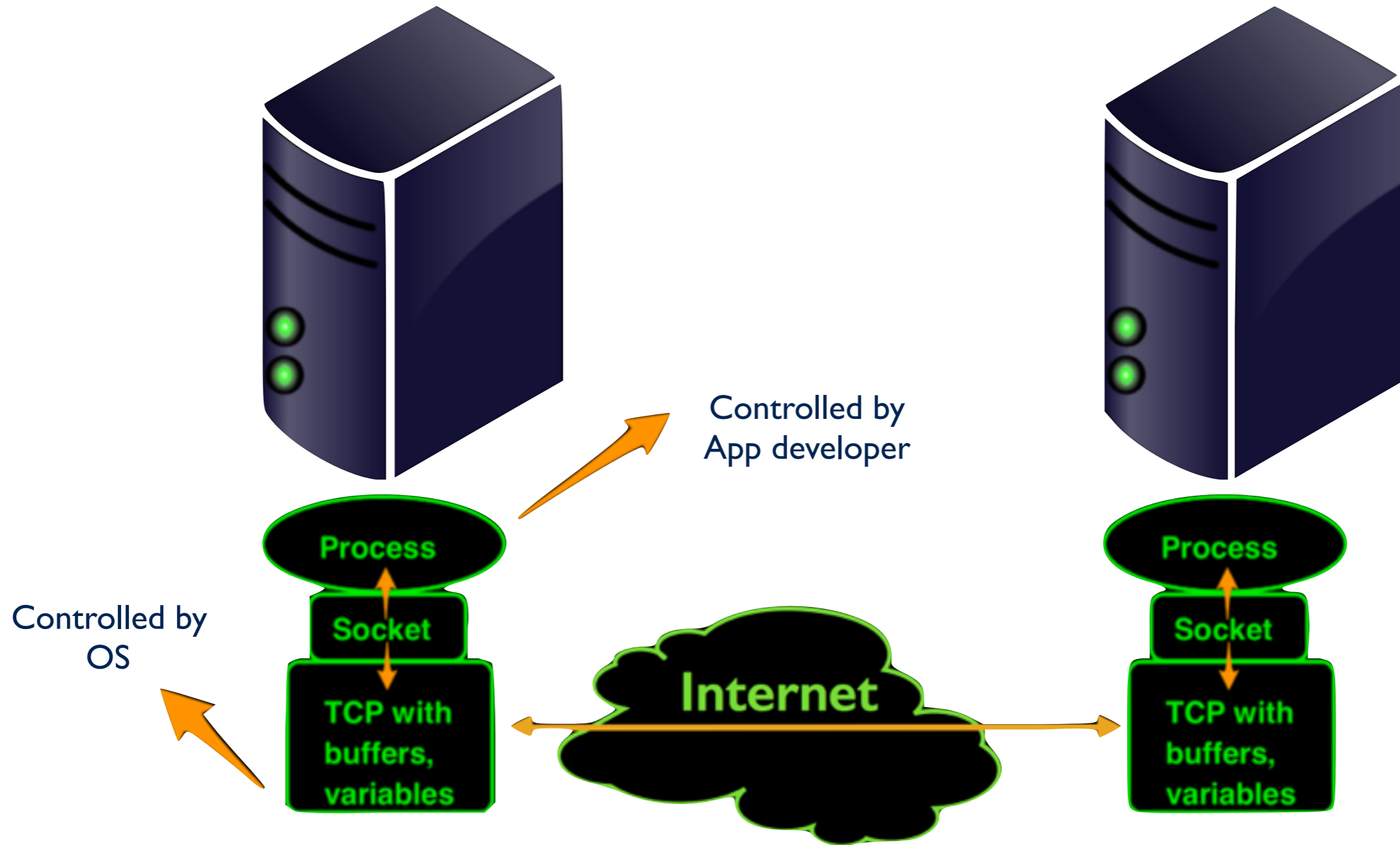# #Layers of the IP Protocol Suite

Application Layer	e.g. http	Application Layer
Transport Layer	e.g. TCP, UDP	Transport Layer
Network Layer	e.g. IP	Network Layer
Link Layer	Ethernet	Link Layer

#Protocol Suite Location

| Application Layer | | Applications (e.g. browser, game, ftp) |

| Transport Layer | | Application Programming Interface (API) (e.g. network API) |

| Network Layer | | Operating System (e.g. Unix) |

| Link Layer | | Interface to the Network Card
Network Card & Device Driver (e.g. Ethernet card) |

#Socket Programming API

Controlled by
App developer

Controlled by
OS

Process

Socket

TCP with
buffers,
variables

Internet

Process

Socket

TCP with
buffers,
variables

#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
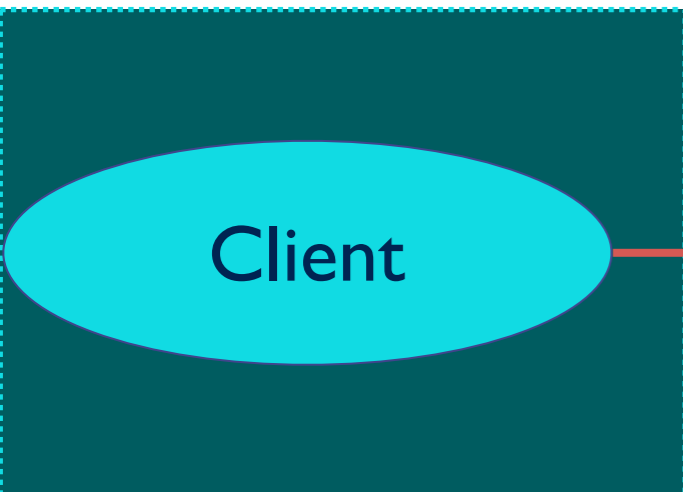
#Client-Server Architecture

response

| Client | | Server |
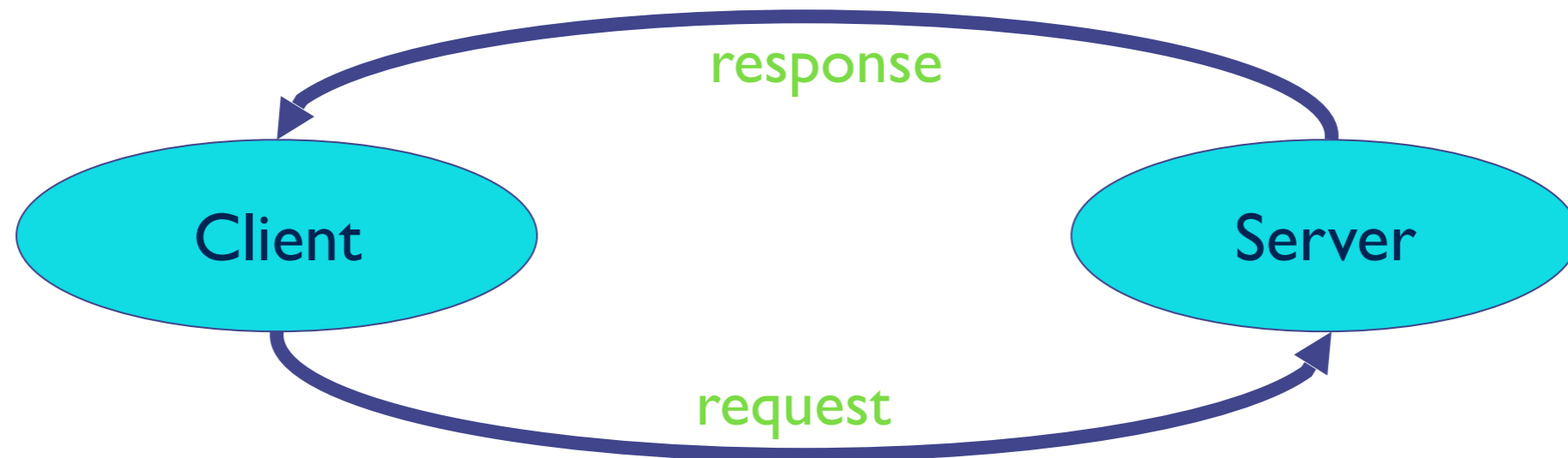
request

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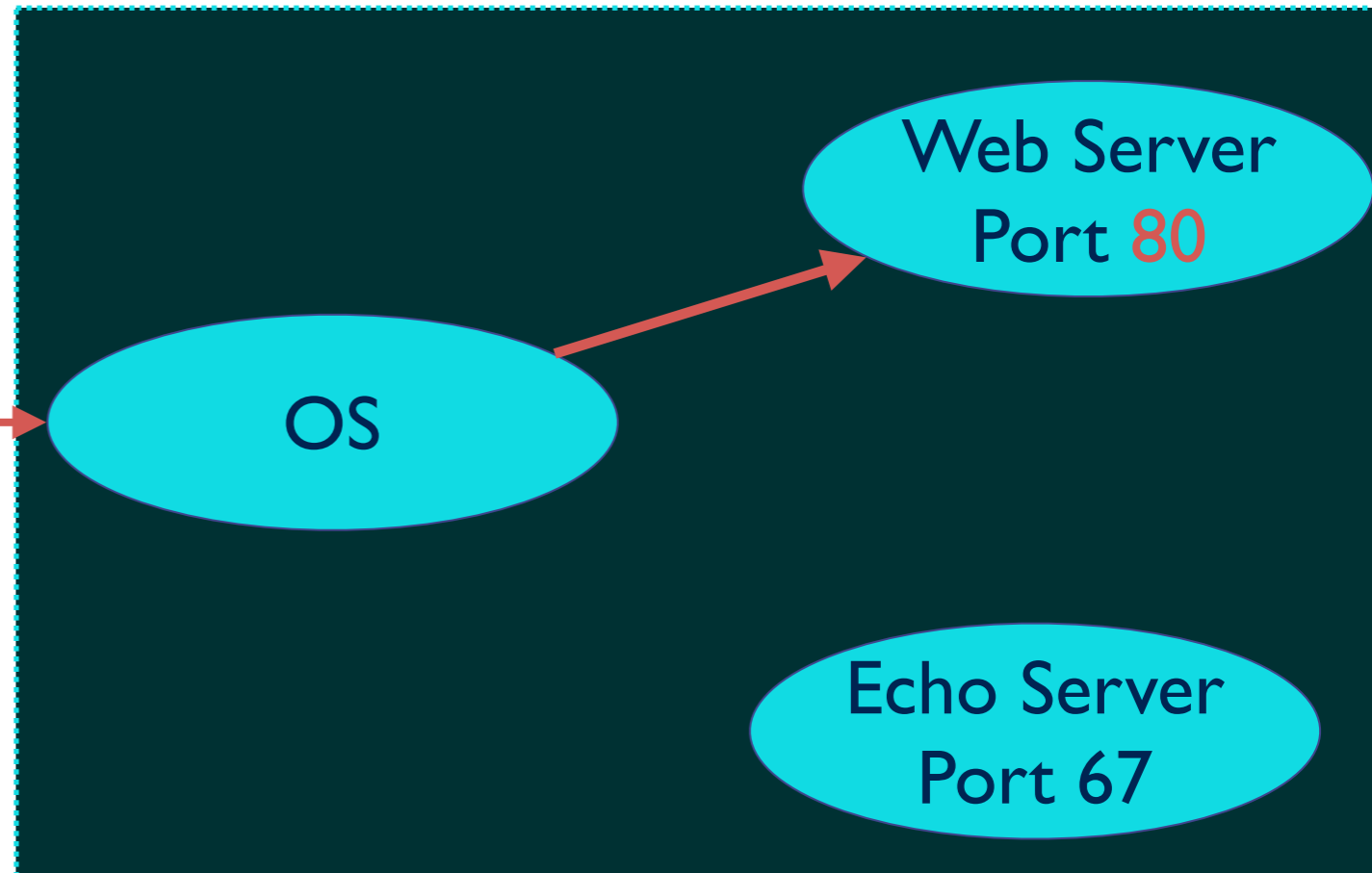- What if Server starts a connection?

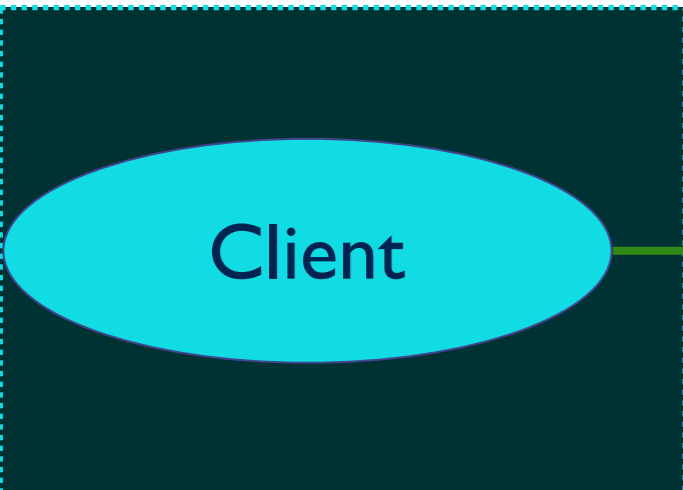# #Identifying the Receiving Process

Server Host 68.90.132.108

Client Host

Client

request for
68.90.132.108:80

OS

Web Server
Port 80

Echo Server
Port 67

#Identifying the Receiving Process

Server Host 68.90.132.108

Client Host

Web Server
Port 80

request for
68.90.132.108:67

Client

OS

Echo Server
Port 67
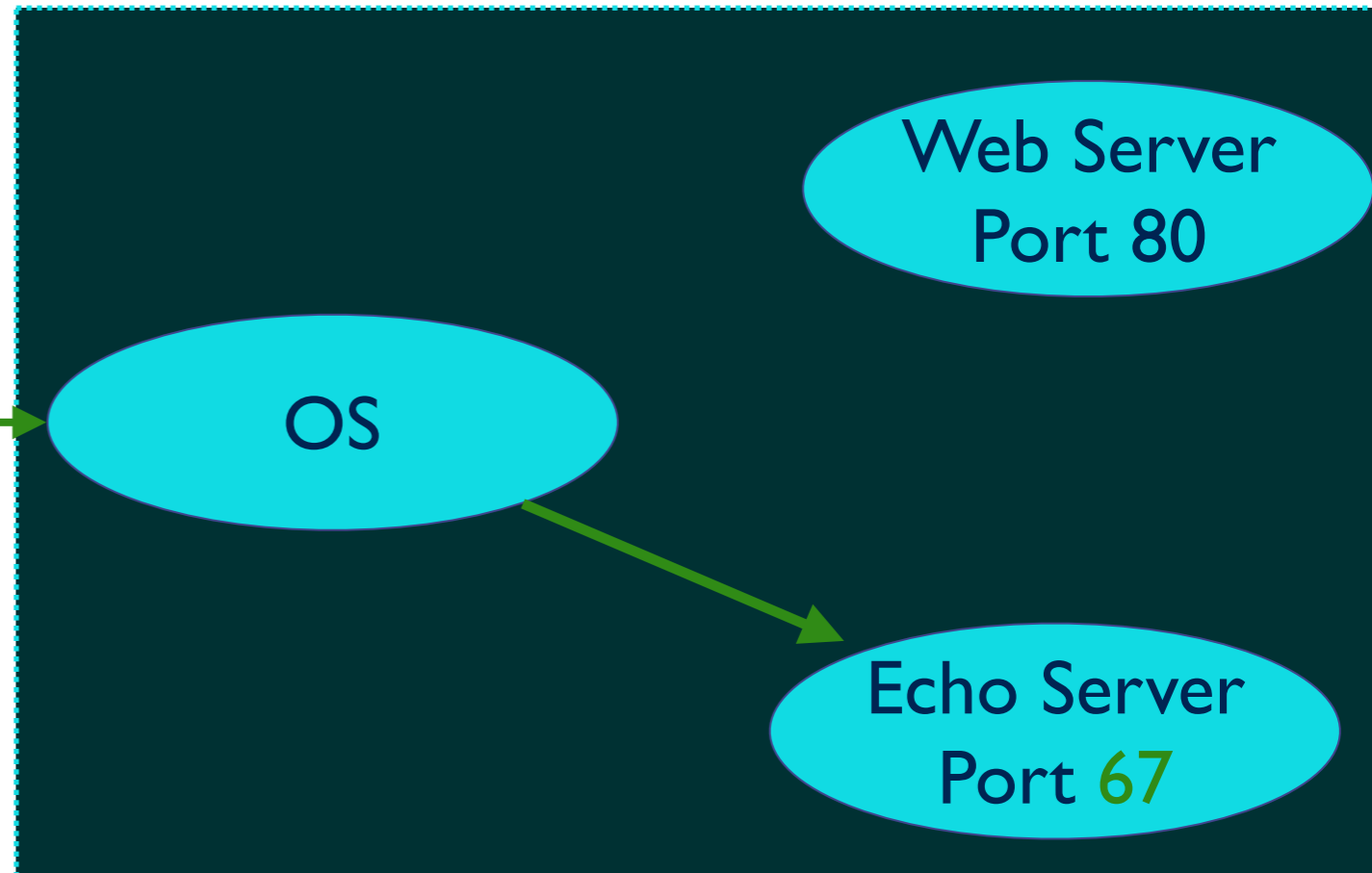
Port Numbers are Unique on Each Host...
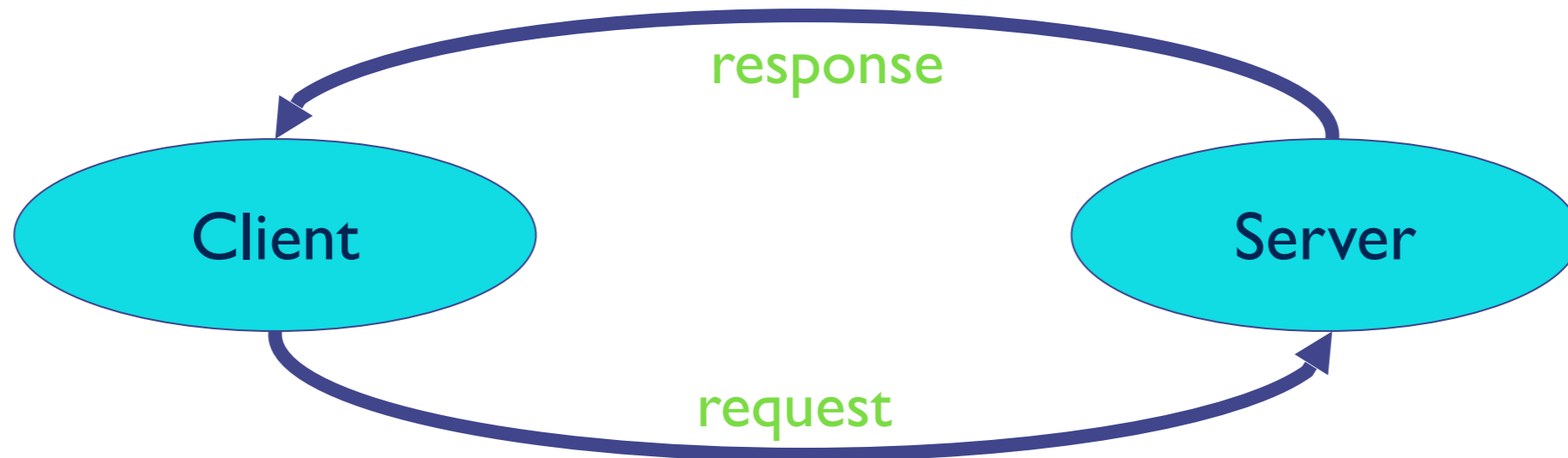
Port number uniquely identifies the socket:

- 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!

response

Client   Server

request

- **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