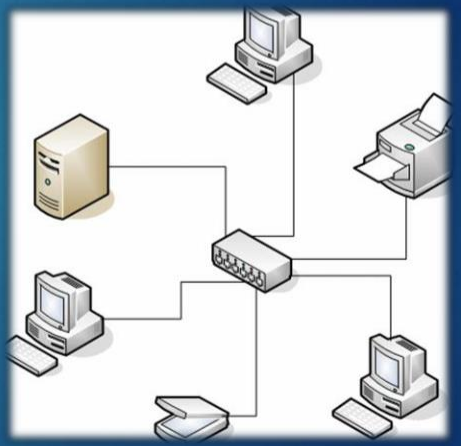
Computer Networks Lab

Week 2: Introduction to Socket Programming

Computer Networks

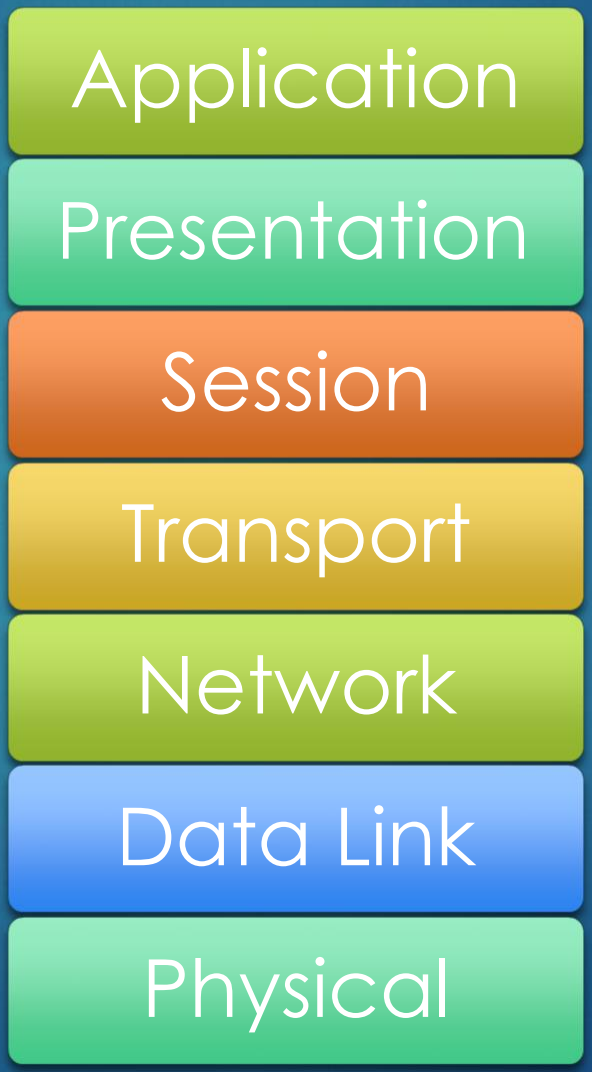
- What is a computer network?
  - Interconnected 2 or more computers or hardware devices
- What is the purpose of Computer networks?
  - Resource sharing
  - Information sharing
  - Communication



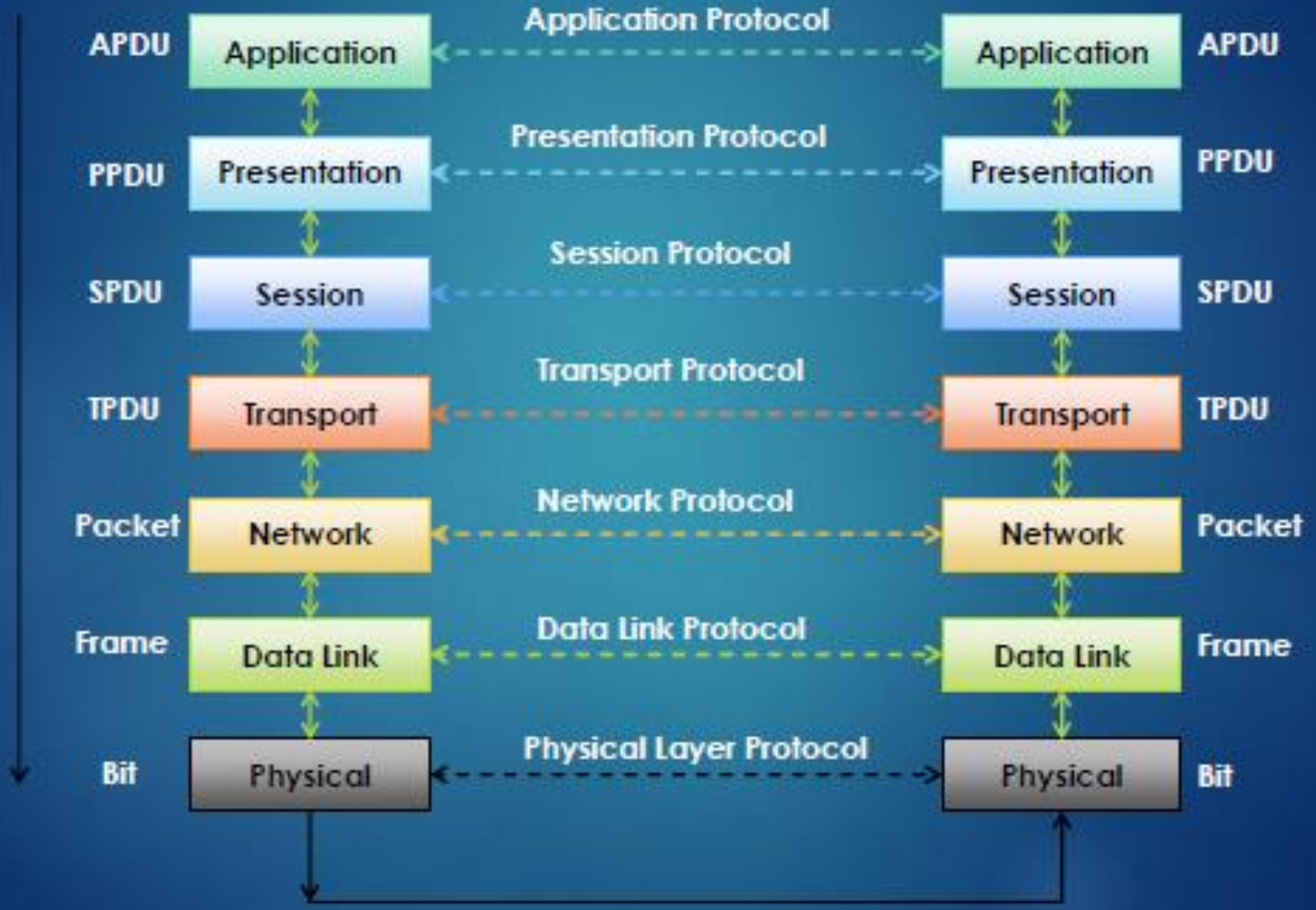
Computer Networks

- PAN (Personal Area Network)
  - 1 m
- LAN (Local Area Network)
  - 10 m to 1000 m (1 km)
- MAN (Metropolitan Area Network)
  - 10 km
- WAN (Wide Area Network)
  - 100 km to 1000 km

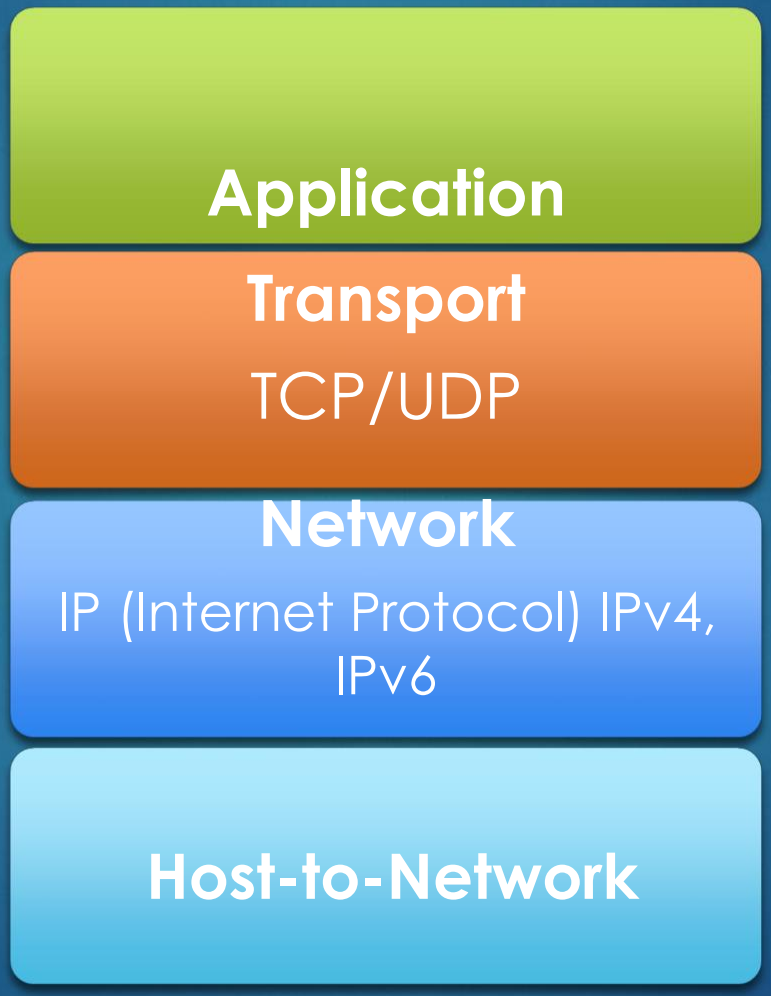
OSI Model


Application
Presentation
Session
Transport
Network
Data Link
Physical

Communication

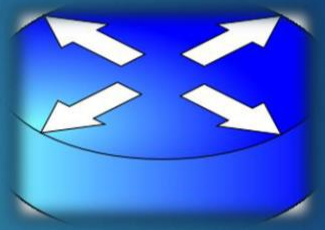


Internet Protocol Suit



Network Devices

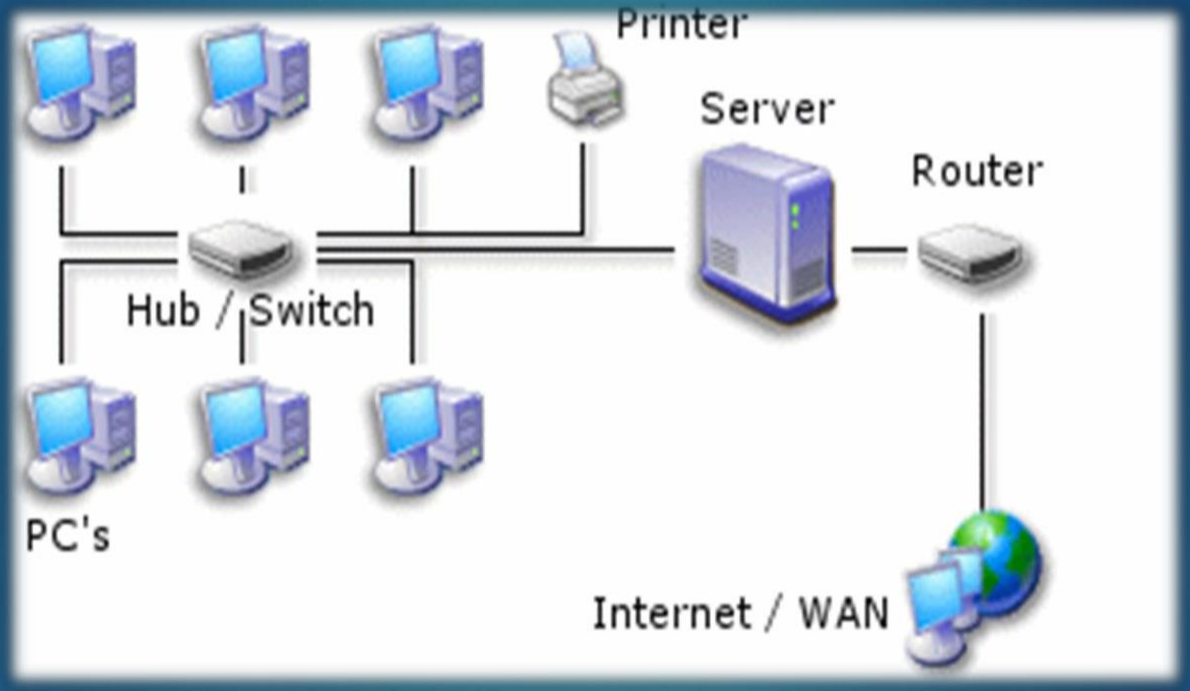
- Routers (Layer 3) Network



- Switch (Layer 2) Data Link

- HUB (Layer 1 device) Physical

Network Devices Use



Addressing

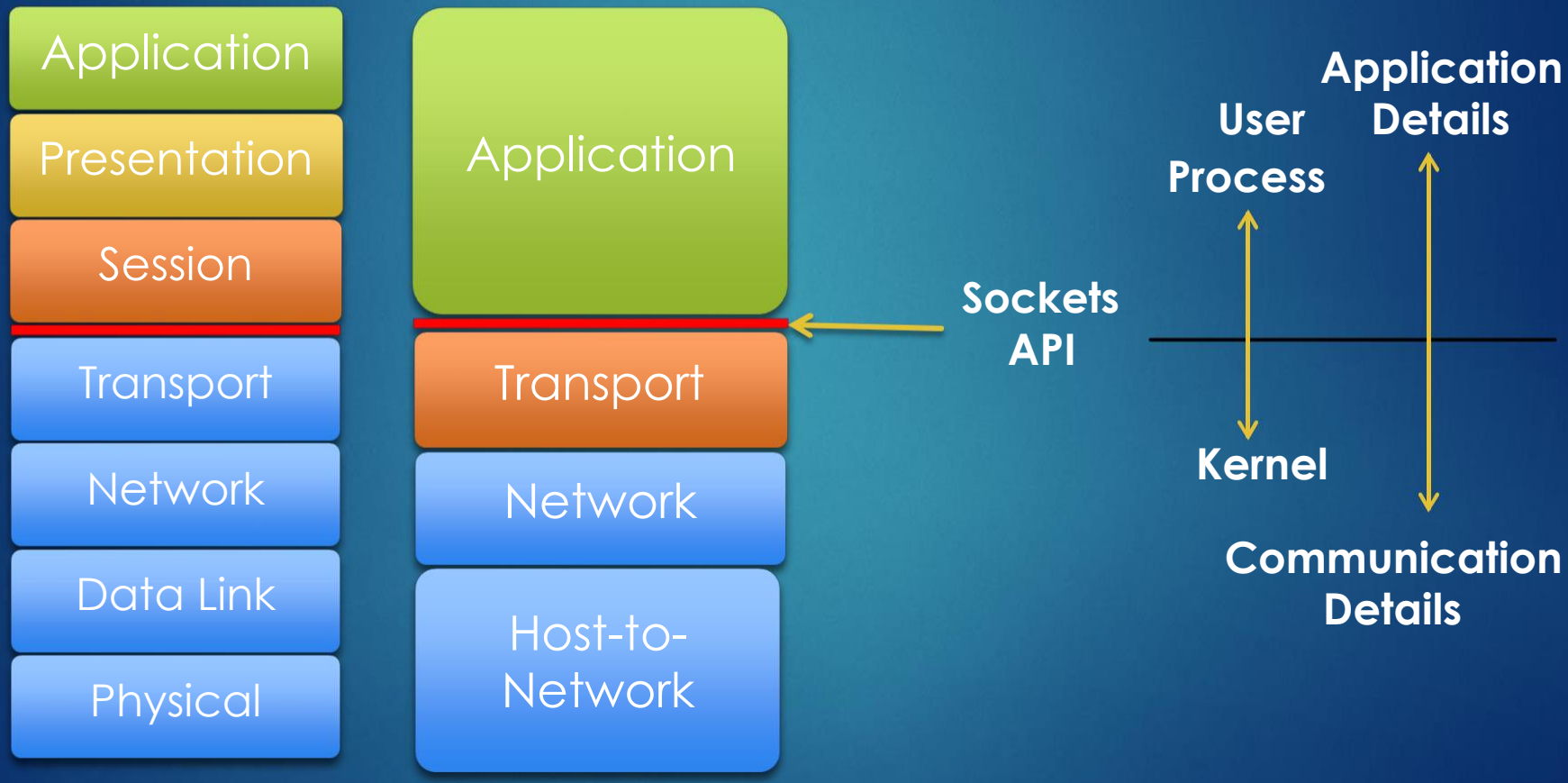
- Mac Address
- IP Address
- Port Address

Socket Programming

- Why Socket Programming?
  - To build any Network Application
  - Web browsers (Internet Explorer , Firefox)
  - Web Apps (Chat, Mail, File Transfer Apps)

What is the Socket?

Socket (An application programming interface(API) for inter process communication)



What is the Socket?

- Socket(Communication End Point)
- Working with Sockets is similar to working with files

File I/O	Socket I/O
Open File	Open Socket
	Name the Socket
	Associate with another Socket
Read and write	Send and Receive between Sockets
Close the File	Close the Socket

- Socket has always an address (**IP and Port**)
- Functionality (Communication)

  One application process can communicate with another application process (local or remote) using a socket.

TCP & UDP

- Difference between UDP and TCP
- Where to use what?
- Applications of UDP
- Applications of TCP

Socket Types

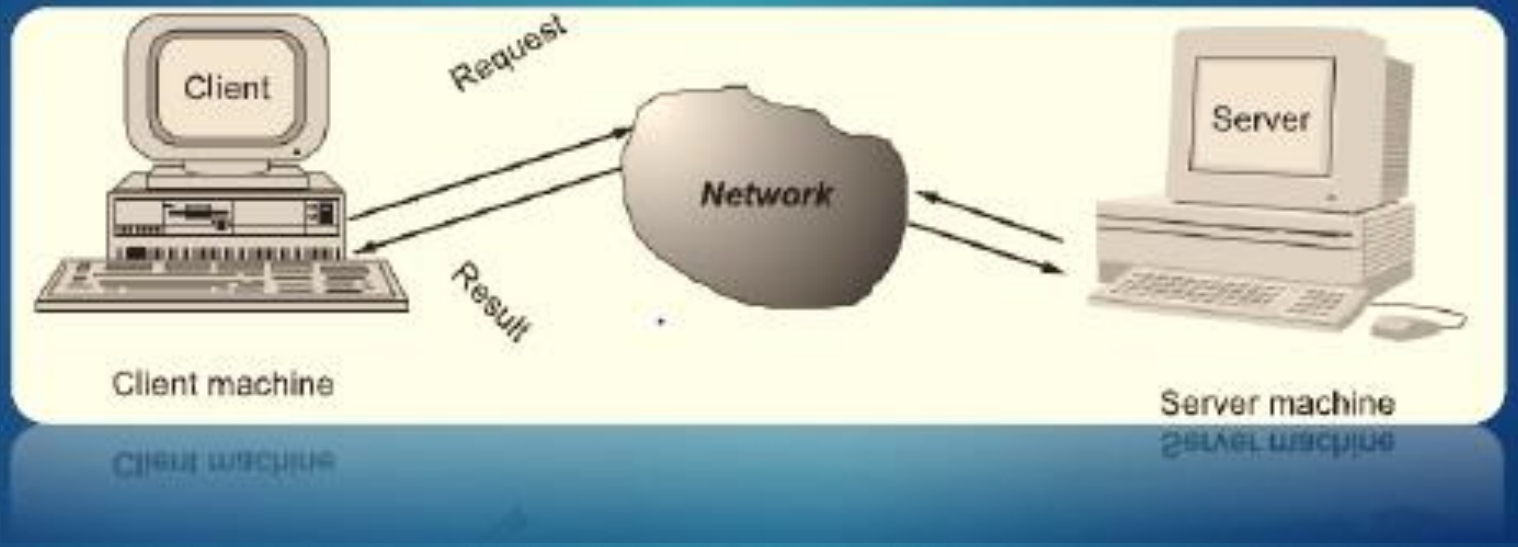
- Stream Sockets (SOCK_STREAM)
  - Connection oriented
  - Rely on TCP to provide reliable two-way connected communication
- Datagram Sockets (SOCK_DGRAM)
  - Rely on UDP
  - Connection is unreliable

Functions used in Socket Programming

Socket()	Endpoint for communication
Bind()	Assign a unique telephone number
Listen()	Wait for a caller
Connect()	Dial a number
Accept()	Receive a call
Send(), Recv()	Talk
Close()	Hang up

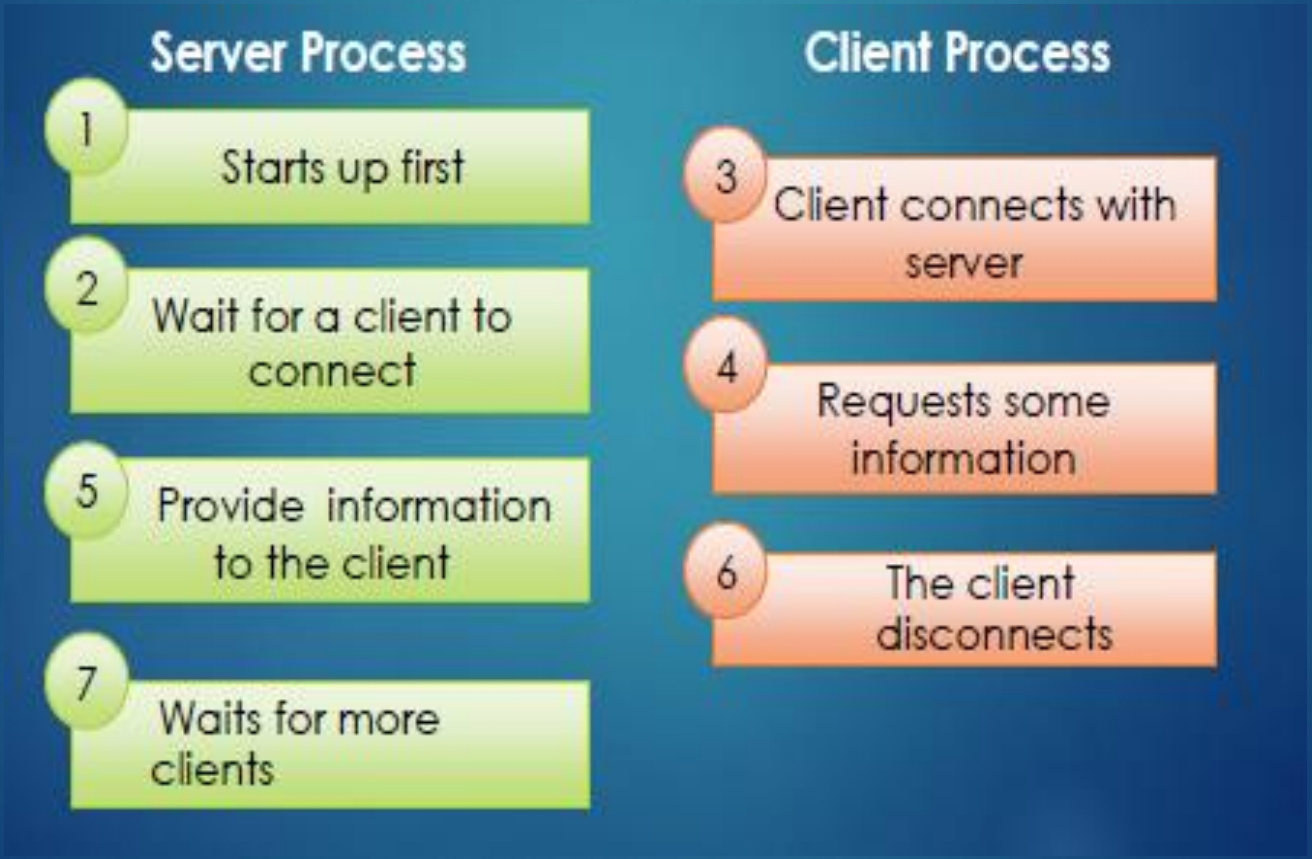
The Client – Server model

- Server – Provider of Services
- Client – Seeker of Services

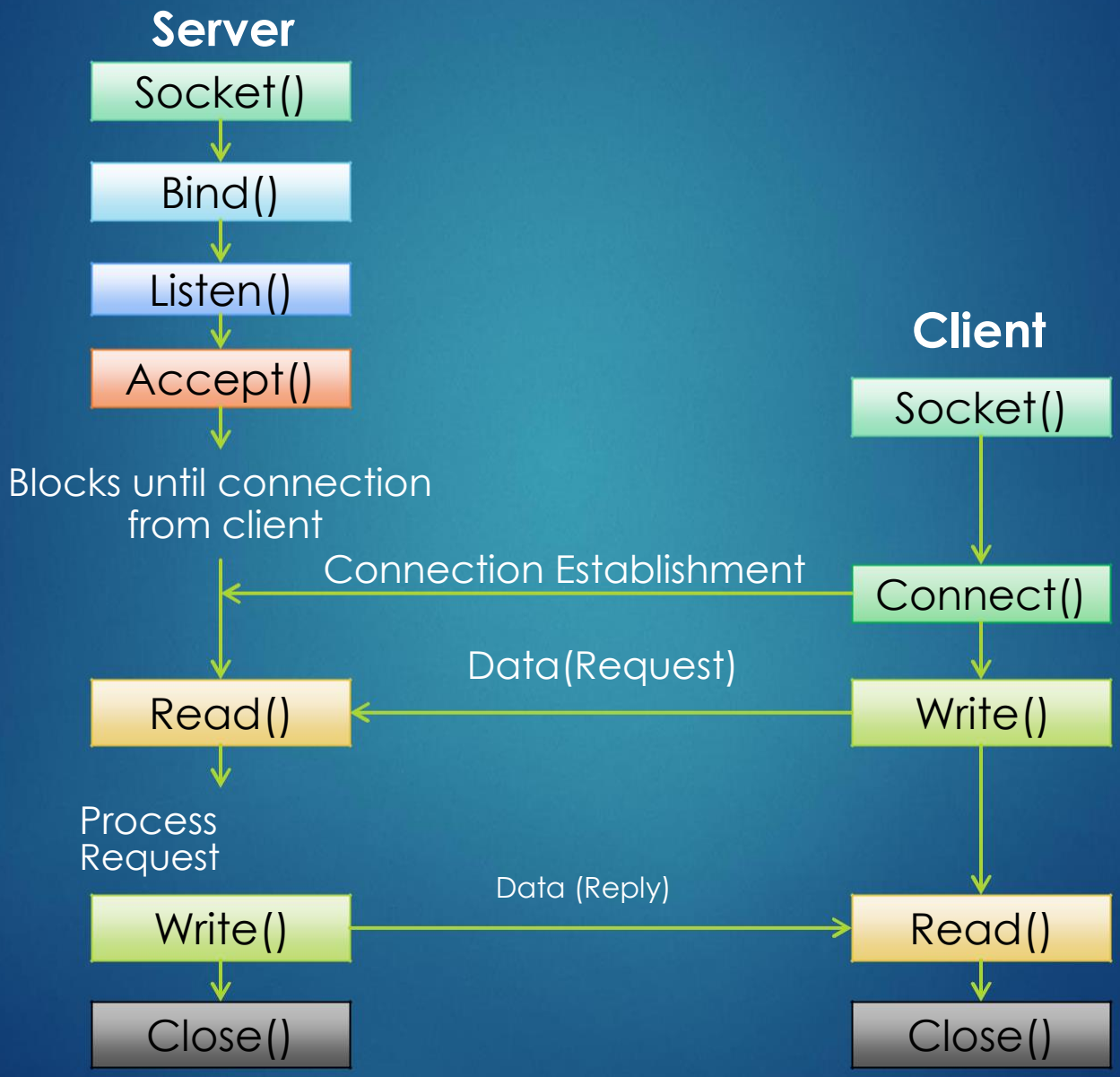


The Client – Server model

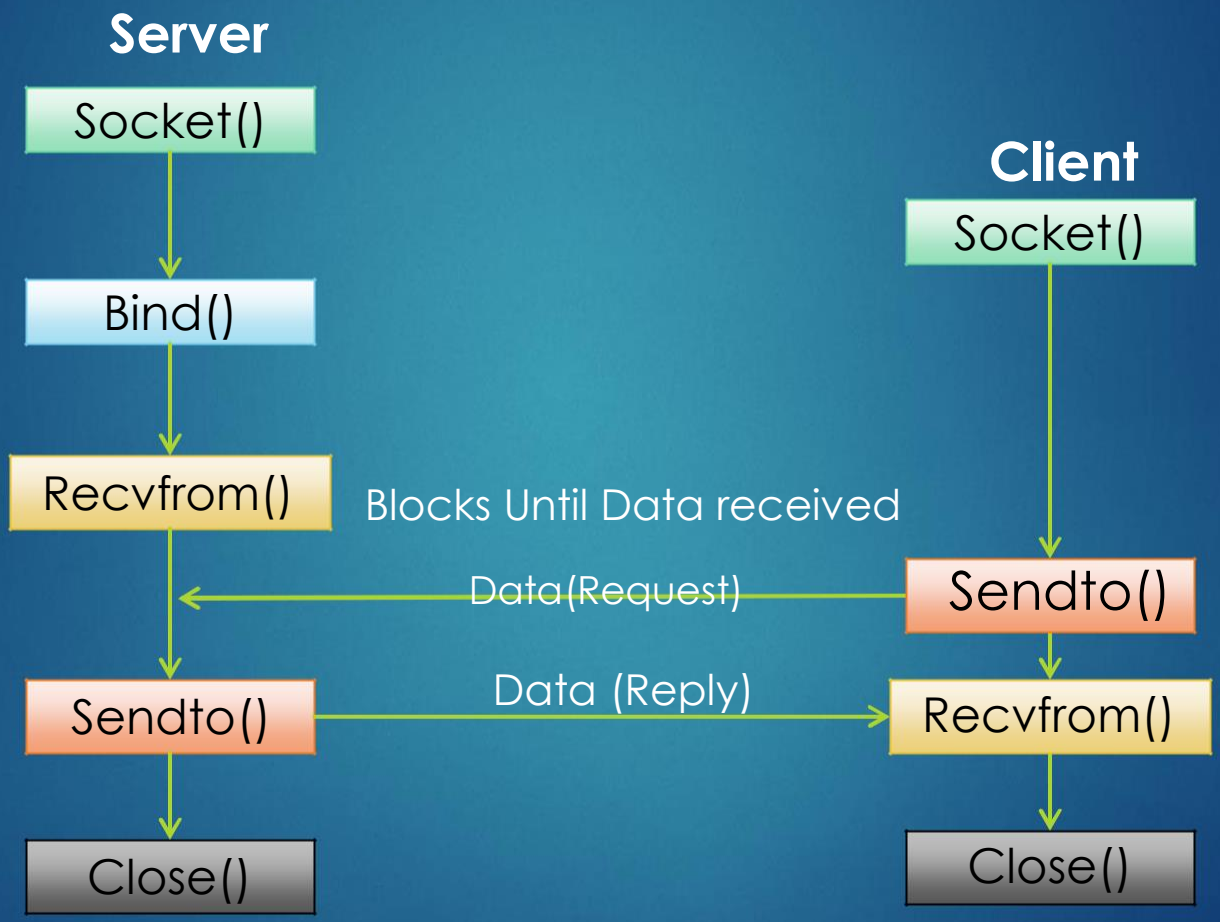
- In the socket programming world almost all communication is based on the Client-Server model.



TCP Server – Client Interaction



UDP Server – Client Interaction



Some Commands

- ipconfig (for IP inquiry) Windows
- Ifconfig (for IP inquiry) Linux
- ipconfig /all to check Mac Address of System

Thank You