NETWORKING BASICS

ESTR2106 2022-23 Term 1

Building Web Applications

Dr. Chuck-jee Chau
chuckjee@cse.cuhk.edu.hk

香港中文大學
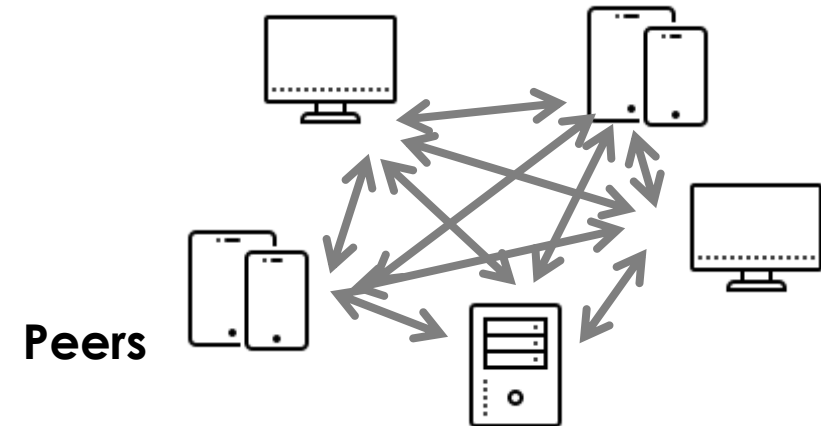The Chinese University of Hong Kong

CLIENT-SERVER ARCHITECTURE

- World Wide Web (WWW) uses the client-server architecture
  - ***Clients*** obtain service from a centralized server
  - ***Server*** waits for client requests and make response
    - e.g., web, email, instant messaging, media streaming, …
  - *See: https://www.britannica.com/technology/client-server-architecture*

Web server

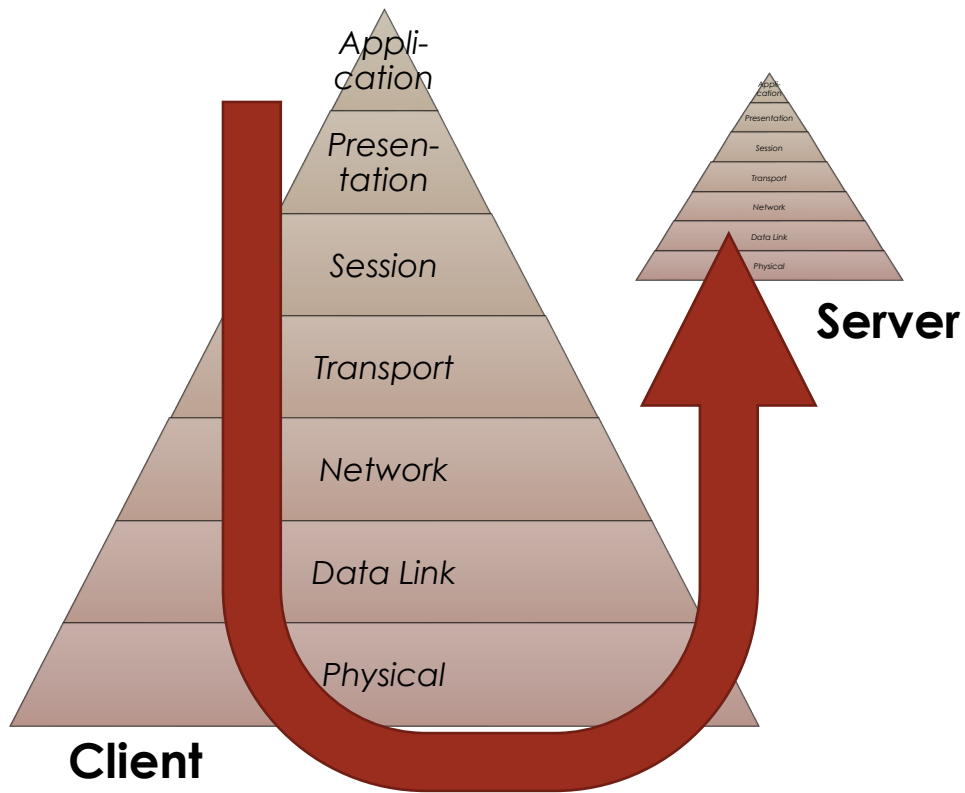**Web clients**　　　　　　　　　　　　　**Web clients**
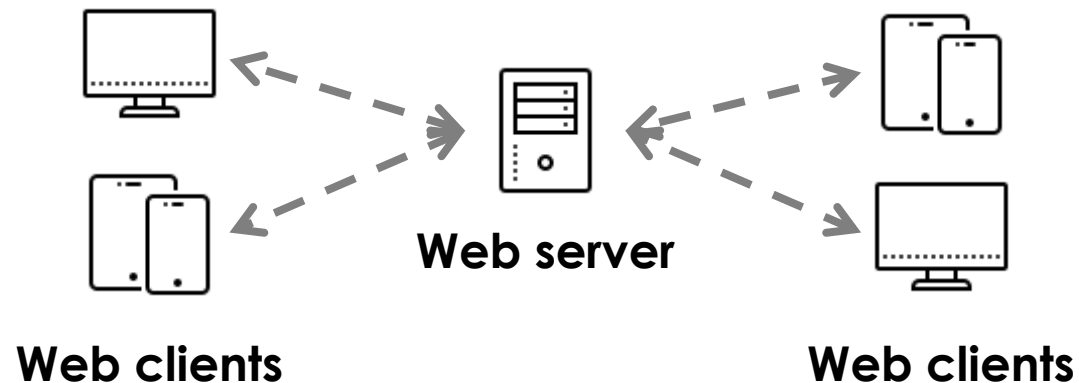
PEER-TO-PEER (P2P) ARCHITECTURE

- Computers or devices are equally privileged participants → both supplier and consumer

- Sharing of tasks, resources and workload
  - e.g., file sharing networks, crypto-currencies, some multimedia applications, …

- *See: https://www.britannica.com/technology/P2P*

Peers

THE OSI NETWORK MODEL

Appli-
cation

Presen-
tation

Session

Transport

Network

Data Link

Physical

Client

Application
Presentation
Session
Transport
Network
Data Link
Physical

Server

- Communication between clients and servers can be seen in ***multiple layers***
  - Abstraction ➔ reducing complexity of problems to smaller ones
  - Division of labour

THE OSI NETWORK MODEL

- Host layers: accurate data delivery between computers
  - Layer 7 "*Application*": facing the user
    - e.g., HTTP, email, games, …
  - Layer 6 "*Presentation*": encoding, data compression, encryption
  - Layer 5 "*Session*": continuous exchange of information
  - Layer 4 "*Transport*": reliable transmission of information
    - e.g., TCP and UDP

THE OSI NETWORK MODEL

- Media layers: ensuring data to arrive at the destination
  - Layer 3 "***Network***": network addressing and routing
    - Note: IP (Internet Protocol) address
  - Layer 2 "***Data Link***": data transmission between two nodes
    - Note: MAC (Medium Access Control) address
  - Layer 1 "***Physical***": raw stream of bits in the physical medium
    - e.g., Ethernet, USB

COMMUNICATION PROTOCOLS

- Clear definition of steps is needed for two computers to communicate
  - Rules
  - Syntax
  - Semantics
  - Synchronization of communication
  - Error recovery methods

- There are protocols for every layer in the networking model

- The **_Internet Engineering Task Force_** (IETF) develops and promotes voluntary Internet standards

MAC AND IP ADDRESSES

- MAC address (*layer 2*): locating a piece of communication device on a local network

- IP address (*layer 3*): identifying a network interface in networks
  - IPv4 (32-bit): e.g., **137.189.91.192** (*www.cse.cuhk.edu.hk*)
  - IPv6 (128-bit): e.g., **2404:6800:4005:808::2004** (*www.google.com*)
  - Public addresses vs private addresses
    - Private addresses limit datagrams to be sent within local network only
    - e.g., 192.168.1.123 is only meaningful within a local network

- The ARP table on a device maintains the correspondence between MAC addresses and IP addresses within a local network
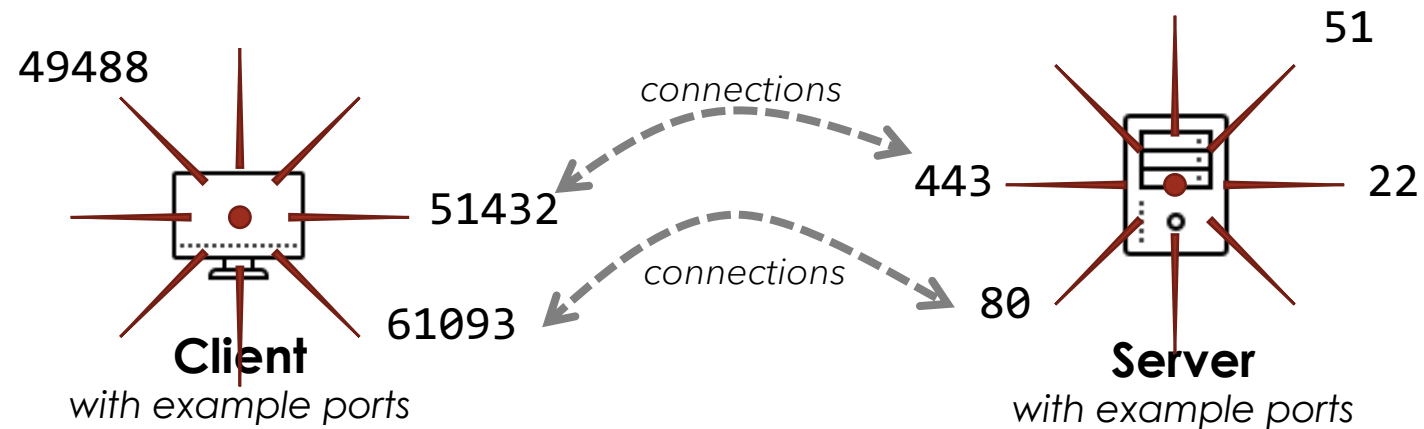
TCP VS. UDP

- Layer 4 protocols: ensuring *reliability* of data transmission

- Transmission Control Protocol (TCP)
  - Handshakes to establish a connection
  - Built-in system to check for errors
  - Guarantee for *data order* and *completeness*
  - Good for: HTTPS, HTTP, etc.

- User Datagram Protocol (UDP)
  - Connectionless ➔ reducing overhead of computation (time!)
  - Extensive error checking and recovery not required
  - Ensuring for *real-time data*
  - Good for: video conferencing, media streaming, etc.

COMMUNICATING OVER PORTS

- In networking, connections are made on ports of a network device

- Each port is "listened to" served by one piece of software (server/client)

  - Well known ports: 0 – 1023 (HTTP: **80**, HTTPS: **443**)

  - Registered ports: 1024 – 49151

  - Private ports: 49152 – 65535

- *See: https://www.webopedia.com/quick_ref/portnumbers.asp*

COMMUNICATING OVER PORTS

Web servers (e.g., Apache) listen on ports *80* and *443*, but you can customize that!

51

49488

connections

The client normally use a random private port for every new connection

51432

443

22

connections

61093

80

Client
with example ports
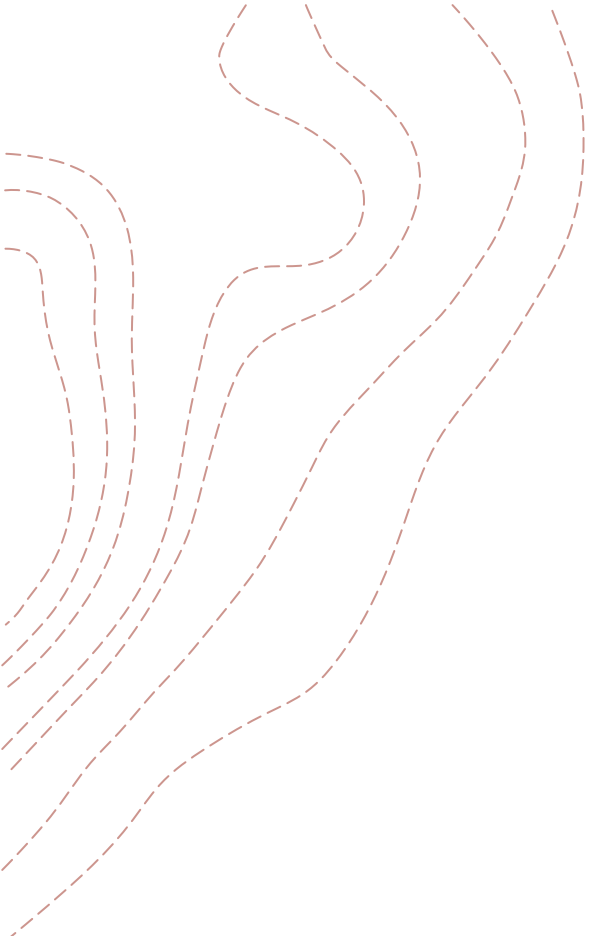
Server
with example ports

SOCKET

- In network programming, a network socket is an endpoint for communication

- Socket = transport protocol + IP address + port number

- Implementation depends on the programming language/environment

LOCALHOST

- It is also possible to have both ends of communication ONLY on one computer

  - Although it is in the same computer, this way allows communication to still go through network layers in the OS

- Usually identified in the computer as "localhost" or 127.0.0.1

connections

3000

Node.js (back-end)

localhost

53323

Chrome (front-end)

READ FURTHER…

The Network Layers Explained

https://www.plixer.com/blog/network-layers-explained/