Computer Network Note

LESSON 18/09/25

Computer Network

1. optional oral, written wajib
    a. no project
    b. Score 14-18 withdraw, below it is insufficient
    c. no neg marking
2. Terms =
    a. Network edge = host (end systems)
        i. Definition = in the border of the network
    b. Network core = exampe: interconnected router in diff countries, network of networks
        i. Definition = Mesh of interconnected routers
        ii. approaches of packet swithing
            1. network forward aka switching= only care abt source n destination
                a. router has a table with header value and output link
                Example = header value: 0100 ; output link: 3
                b. can dynamically update path
                c. two actions =
                    i. Local action = input to output
                    ii. Global action = find the shortest path
                d. Links n roouter
    c. internet protocol = http, tcp, ip, ethernet
        i. definition = the format order of message send and received among netwrok entities and action taken on message transmission
        ii. Steps = TC connection sent, request, get(),
    d. performance = loss, delay, throughput (amount of data that can be transmitted)
    e. Security
    f. Packet switching = router, switches
        i. Steps
            1. Takes message tbd
            2. breaks into smaller packets, of length L = bits
            3. trasmit at transmission rate R, where R = bits/sec
                a. link transmission rate as link bandwith
        ii. Formula
            1. Packet transmission delay = time needed to transmit l bit packet to link == L / R
        iii. Store and forward = packet must arrive at router as a whole b4 transmitted
        iv. Packet queueing and loss = if arrival rate excee transmission rate, there ll be queue, packet will be loss if memory filled
        v. Adv n diasdv = simple but can have congestion
        vi. Packet structure

1. Header
2. Target
3. Sender

g. Circuit switching
    i. end-end resource allocated for transmission, where the resource is dedicated for a single activity
        1. NO data congestion will happen
        2. but not many data can be processed
h. Packet switching vs circuit switching
    i. Packet switching more user, but with but slower speed / higher prob of failing or congention
i. Communication inks
j. Network types
k. Infrastructure
l. Microprocessor = chip without OS, use firmware instead, used on fridge without internet
m. Internet standard = RFC (request for comment), IETF (internet engineering task force)
n. Programming interface = hooks allowed apps to connect to internet
o. Access network =
    i. Cable based access =
        1. frequency division multiplexing: diff channel transmitted
            a. Optical electromagnetic divided into narrow freq bands
        2. Time division multiplexing
            a. Time divided into slots
        3. Hybrid fiber coax: asymmetric
    ii. Digital subscriber line
    iii. Home network = ,
        1. modem, router, wired ethernet, wifi access point = combined to a single box
        2. Types
            a. Wireless local area network
            b. Wide area cellular access network
            c. Enterprise network
            d. Data center network
p. Physical media
    i. Directed, undirected
    ii. Cable
        1. Coaxial cable = two copper conductor
        2. fiber optic cable = glass fiber carrying light pulse
    iii. Wireless = use electromagnetic
        1. Radio link types
            a. Wireless LAN
            b. wide area

        c. Bluetooth
        d. Terrestrial microwave
        e. Satellite

q. Internet service provider =
    i. host connect internet via access internet service protocol
    ii. access ISP will be connected
    iii. how isp are connected
        1. local isp willl be connected to regional isp
        2. ISP will be interconnected with a global transit ISP
        3. Glbal transit point will be connected by internet exchange point using a peering link
    iv. Cache data center used to reduce need to visit global isp
        1. move closer to reduce propagation delay
r. Packet delay and loss
    i. packet loss when memory full
    ii. Memory known as buffer
    iii. Stage =
        1. Transmission
        2. Propagation == when its on the way
        3. Nodal processing = check bit error, determine output link
        4. queueing
    iv. Formula = d nodal = d proc + dqueue + dtrans + dprop
        1. d trans = L / R
        2. d prop = d / s
            a. d = length of physical link
            b. s = propagation speed
        3. d queue is ignored bcs very small
    v. Example question
        1. 10 package, dtrans = 12 second, d = 100 km, s = 100 km/hr, how much till queue on the second gate
    vi. Variables
        1. a = avg packet arrival rate
        2. L = packet length
        3. R = bit transmission rate
        4. L A / R = traffic density
            a. if 0 = no delay
            b. if more than one = traffic queue incoming, delat infinite
            c. 0 to 1 = delay is large
        5. Throughput = bits / time unit
    vii. Traceroute format = source pack1 pack2 pack3 to target pack1 pack2 pack3 ;; uses miliseconds ;
        1. delays may increase or decrease
        2. if no time == packet loss
    viii. throughput

1. bottleneck link = contstains in end-end throughput
2. Bigger capacity on data center (Rs) than on ppersonal computer (Rc)

ix. Security
1. Cybercrime type
    a. packet sniffing = intercept the packet passing by
    b. IP spoofing = injecting a packet with false source address
    c. Denial of service = attacer make resource unavailable by overwheliming resource with bogus traffic
2. Defense = authentication, confidentality, integrity check, access restriction, firewall

x. Protocol layer

LIST of TCP/IP/Transportation Control Protocol layers

1. Application
2. Transport
3. Internet
4. Network Access

OR the seven layer model called **INTERNET PROTOCOL STACK/ISO**

1. Application
2. Presentation
3. Session
4. Transport
5. Network
6. Link
7. Physical

5 layer Subject model

1. Application
2. Transportlayer
3. network
4. link
5. physical

LESSON 25/09/25

Theme = application layer

1. Client server paradigm
    a. Server
        i. Always on host
        ii. Permanent IP address
        iii. used in data center for scaling
            1. Vertical scaling = increase ram or cpu
            2. Horizontal scaling = increase number of machine

    b. Clients (one server has many clients but one client has one server)
        i. contact with server
        ii. may be intermittently connected
        iii. have dynamic ip address
        iv. dont communicate with each other
        v. example = http, imap, ftp
2. Peer to peer architecture
    a. no always on server
        i. arbitrary end systems directly communicate
        ii. self scalability = new peer bring new service capacity as well as new service demands
        iii. example = bittorent
3. Processes communicating
    a. Process = program running within a host
    b. within same host, two processes communicate with inter processes
4. Socket = process send/receive message to/from its socket
    a. Located between app process and transport layer
5. Addressing processes
    a. host device has unique 32 bit ip address
    b. to receive messages, process must have identifier
        i. identifier contain ip address and port numbers associated with process on host
    c. Example = http server 80
6. What application layer protocol define
    a. types of message exchanged
    b. Message syntax
    c. message semantics
    d. rules
    e. open protocols
    f. proprietary protocol
7. Transport service needed
    a. Data integrity (some apps require reliability), timing (low delay), throughput (min throughput value to run), security

Theme new = transport layer

1. TCP service
    a. ensure reliable transport, flow and congestion control, connection oriented, no timing provided
2. UDP service
    a. Unreliable data transfer, and doesn't provide flow control and reliability
3. Securing TCP methods
    a. Vanilla TCP and UDP socket (No encryption at all)
    b. Transport layer security (TLS) = provide encryption, data integrity, and end point authentication

4. Web and HTTP
    a. webpage = a collection of digital objects each can be stored on different web servers – example: database server give content to frontend server
        i. consist of base html file consisting several referenced objects
        ii. example =
            1. www.someschool.edu = host name
            2. someDept/pic.gif = path name
    b. HTTP = hypertext transfer protocol using client/server mode ; client request, recieve, and display web object ; server send objects
        i. HTTP use TCP = initiated by client (or create socket)
        ii. after http message are exchanged, close TCP connection
        iii. Stateless http = contain no info on past client request
        iv. Types of http
            1. Non persistent = using tcp connnection, at most one sìobject sent, kirim packet satu-satu
                a. Disadv = has RTT time delay
            2. Persistent = using tcp connection, send mulitple objects
        v. Steps = initiate tcp connection, request file, transmit file, file recieved ; non persistent will have two rtts, while persistent http have only 1 rtp
    c. HTTP request message structure =
        i. Request line = example == get, post, head
            1. method-URL-version-lf
        ii. Header lines
            1. HeaderFieldName-Value-lf
        iii. Body
            1. entitybody
        iv. carriage return
    d. Other http request message
        i. Post = Create an object in the server
        ii. Head = request head that would be returned if specified
        iii. Get = Read user data in th url
        iv. Put = update new files into the server
    e. HTTP codes = 200 (ok), 301 (move permanently), 400 bad request (not understood by server), 404 not found (req doc not fund, 505 (version not supported)
5. maintaining user/server state with cookies :
    a. stateful protocol = full change or none at all
    b. Components of cookies
        i. header line of http response mesage
        ii. —”---- —--- in the next http request message
        iii. cookie file kept on user host, managde by user browser
        iv. back end db at website
    c. Cookies definition = maintain some state bween transactions

        i. example = store recommendation algorithm, maintain state for autologout when inactive
        ii. Third party cookie = cookie fro website not chosen to be visited
6. Web caches = satisfy cient request without involving origin server
    a. download wireshark HW

LESSON 26/09/25 – self study

A. Target
    a. Catch up todays lesson with PDF
    b. Missed lectures =>
        i. Introduction part B
        ii. Wireshark introduction

B. Slides Review one

1. Contents
    a. ▪ ISO/OSI Reference Model
    b. ▪ Transmission Modes
    c. ▪ Line Configuration
    d. ▪ Unicast, Broadcast and Multicast
    e. ▪ Topology
    f. ▪ Types of Area Network
    g. ▪ Networking Trends
    h. Internet of Everything

2. ISO / OSI reference model = the two layers that doesn't exist in **Internet Protocol Stack** yet is important
    a. Presentation = allow app to interpret meaning of data, example = encryption
    b. Session = act as checkpoint/synchronization/recovery for data exchange recovery
3. TCP/IP vs ISO/OSI layer
    a. TCP uses four layer only, osi use 7

TCP/IP	OSI
TCP refers to Transmission Control Protocol.	OSI refers to Open Systems Interconnection.
TCP/IP has 4 layers.	OSI has 7 layers.
TCP/IP is more reliable	OSI is less reliable
TCP/IP does not have very strict boundaries.	OSI has strict boundaries
TCP/IP follow a horizontal approach.	OSI follows a vertical approach.
TCP/IP uses both session and presentation layer in the application layer itself.	OSI uses different session and presentation layers.
TCP/IP developed protocols then model.	OSI developed model then protocol.
Transport layer in TCP/IP does not provide assurance delivery of packets.	In OSI model, transport layer provides assurance delivery of packets.
TCP/IP model network layer only provides connection less services.	Connection less and connection oriented both services are provided by network layer in OSI model.
Protocols cannot be replaced easily in TCP/IP model.	While in OSI model, Protocols are better covered and is easy to replace with the change in technology.

   b.
   c. OSI is less reliable, use model then protocol, has stricter boundaries
   d. TCP is more reliable, use protocol then model, less strict boundaries
4. Transmission mode = transferring data through devices with bus and networks, <u>types including =</u>
   a. Simplex =
      i. uni-directional (one way),
      ii. only one of the device can transmit and the other can only receive,
      iii. example = laptop and keyboard
   b. Half-duplex =
      i. both station can both transmit and receive, but not at the same time
      ii. Entire capacity used for each direction;
      iii. capacity = bandwidth x propagation delay
      iv. Example = walkie talkie
   c. Full duplex
      i. Both directions both roles, at the same time
      ii. Capacity is divided between signals
      iii. capacity = 2 x bandwidth x propagation delay
5. Line Configuration = two or more devices connected through a link
   a. Link = communication pathway that transfer data
6. Point to point connection = dedicated link between two devices
7. Multipoint connection / Multidrop configuration = when two or more devices share a single link, capacity is shared between devices; two possible configuration:
   a. Spatial sharing = several device share link simultaneously
   b. Temporal sharing = when user take turn to use the link
8. Unicast, broadcast, multicast difference

LESSON 9 October 2025

1. CRUD = create (post), read (get), update (put), delete (delete)
    a. Use bottle, request, cURL
    b. form standard =
        i. localhost:5000/products/
    c. Bottle = simple fast lightweight micro web framework
        i. from bottle import route, run, template
        @route('/'hello/name')
        def index(name):
            return template(, name=name)
        run(host='')
    d. pip = package manager
    e. virtual environment = python library that isolate application, so that when updating the module of an app, the old version retains the old version and keep on running
        i. make a folder that contains virtual env
        ii. python3 -virtualenv -p python3 venv
    f. Terminal terms
        i. use '%ls' to visualize a folder
        ii. 'cd' change directory
        iii. 'pip freeze' to see all version numb of every installed library and see whether virtual env is active
        iv. 'source venv/bin/activate' to activate venv
        v. 'pip install bottle' to instal bottle
        vi. import things from bottle
            1. from bottle import run
            2. from bottle import post, get, update, delete
        vii. host = 'localhost' to specify localhost
        viii. port = 5000 to specify port
        ix. turn a dictionary data type into a json = return json.dumps(dictName)
        x. 'deactivate' to remove bottle
    g. Localhost is automatically on when laptop is turned on
        i. to enter a localhost, do 'localhost:8000'
    h. To stop web server = ctrl+c
    i. using "__main__":
        i. if __name__ == "__main__"
            main()
        ii. if the file executed is not the code file name, it will not execute
    j. 'return' without any parameter will return code 200 if succeed
        i. Always return code in every case

LESSON 10 October 2025

a. Application to do crud operation = postman
b. POST = Create ; GET = Read ; PUT = Update ; DELETE = delete
c. Github link = https://github.com/lcarnevale/http-examples
d. Make request HTTP
    i. r = request.get('https://", auth= ('user', 'pass')
    ii. Pip freeze = see all pip package versions
    iii. Dictionary = string, json = dict
e. json.dumps() converts from a dictionary to a json file
f. Architecture concluded
    i. Two main elements – server and client
        1. Server must be on at all time,
        2. has two address id
            a. Localhost (road number)
            b. Port (specific house number)
        3. Communication between server and client is done on another layer (not application layer)
        4. Client location is not needed
        5. Client send request and server send response
        6. Server has post, get, put, delete

LESSON 23 October 2025 – appllication layer and transport layer

I. Unit one = application layer

1. Email =
    a. major components = user agent, mail server, simple mail transfer protocol
        i. User agent == mail reader
            1. Compose letters
        ii. Mail server == mailbox containing incoming messages for user
            1. incoming = mailbox, message queue = outgoing messages
        iii. SMTP protocol = bween mail server to send email messages
            1. Client = send email to server
        iv. SMTP RFC = uses TCP to reliably transfer email message from client to server with port 25
            1. client initiate TCP connection
            2. communication bween clent and server done through RTT
            3. after TCP connection initiated, three phases of transfer is done
                a. SMTP handshaking is done
                b. after that, SMTP transfers is done
                c. after that.. smtp closure
        v. Example scenario of email sending
            1. Client uses User agent to compose email to target

2. client's UA send message to her mail server using SMTP, message is located message queue
3. client side of SMTP opens TCP connection with target mail sever
4. SMTP client send message over TCP connection
5. Target mail server place message in target mailbox
6. Target uses his user agent to read the message

vi. SMTP != HTTP, whereas SMTP do client push while HTTP does client pull ; both have ASCI command response
1. HTTP = each object has its own response message
2. SMTP =
    a. multiple objects sent in multipart message
    b. uses persistent connections
    c. require message header and body
    d. uses CRLF
3. SMTP sending mail format
    a. Header = to from
    b. body

vii. reading mail format
1. SMTP = delivery/storage of email message to receiver server
2. IMAP = mail acces protofol
3. HTTP = provides web based on top of SMTP

b. DNS = domain name system

i. definition = a distributed database implemented in hierarchy of many name servers

ii. implemented inside application layer protocol = host, DNS server communicate to resolve names

iii. Servces
1. Hostname to IP address
2. host aliasing = canonical (real) / alias names
3. Load distribution = balance workload

iv. DNS is not centralized but is ditributed because single point of failure, traffic volume, and maintenance and DNS cannot scale (a single server cant scale while a distributed one can)

v. Distributed DNS == distributed database with more reads than writes, organizationally and physically decentralized

vi. Hierarchy of DNS =
1. Root DNS server = internet cannot function without DNS root server
2. top level (.org, .com)
3. authoritative (amazon.org, yahoo.com)

vii. Local DNS name servers = when host make DNS qeuery, its sent to its local DNS server, whereas Local dns server returns reply
1. each ISP (internet service provider) have their own local dns name server

2. when requesting IP address end user will communicate with Local DNS first before going through root, tld, and authoritative then back to end user
    a. done when first time entering a website

viii. DNS protocol message
1. Identification
2. question
3. answer
4. authority
5. additional info

ix. Type mx = for email server , type A = for website

x. DNS security
1. attack type 1 = DDOS attack
    a. Bombard root servers with traffic or bombard tld server
2. Spoofing attack
    a. intercept dns queries and returning false (hoax) replies

2. Peer to peer architecture
    a. no server or client used, not always-on, peers are intermittently connected
    b. example = bittorrent
    c. time to send one copy = F/Us whereas F = file size, Us = upload capacity of server
    d. in client-server, the higher number of client, the longer the time
    e. in p2p, each peer contribute with upload time thus less time increase
    f. Bit torrent implementation =
        i. file divided into chunks
        ii. peers in torrent send/receive file chunks
        iii. each peer exchange with each other, as diff peer have diff subsets of chunks
        iv. each chunks are requested from all peer starting rom the rarest

3. Video streaming and CDNs =
    a. challenge =
        i. heterogeneity (different capabilities background of different users
        ii. video bandwidth
        iii. continuous playout constraint
        iv. user interactivity
    b. throughput example = 24 images / sec
    c. two types of coding video = spatial (info about current image), temporal (info difference about different bween one image to another)
    d. cbr = constant bit rate (fixed encoding rate)
    e. vbr = variable bit rate (video encoding rate that change with spatial temporal coding)
    f. example = server - internet – client
    g. rate video transmission - client video reception, rate video playout at client

h. DASH = dynamic adaptive streaming over HTTP == a system for multimedia streaming
    i. Server = divide into chunks with their own URL (manifest file)
    ii. Client = estimate bandwidth, chooses max coding rate (decreasing quality in exchange
        1. Task = when to request chunk , what encoding rate where to request chunk
    iii. choose different coding rate form diff server
i. Content distribution network = livestreaming to millions of people
j. Example Netflix implementation
    i. end user request content, service provider return manifest
k. OTT = over the top

II. Unit two transport layer

a. TCP (transmission control protocol) , UDP (user datagram protocol) layer
    i. TCP is reliable
    ii. UDP is not reliable unordered but is used on things like videos (millisecond dont really matter)
b. TCP
    i. Transport protocol actions
        1. Dender breaks into packets
    ii. Example = kids in the house send letter to kids in another house
        1. House = host
        2. kids = processes
        3. app messages = letters
    iii. Core info
        1. Process to process relation is part of network layer
        2. Host to host is transportation layer
        3. writing the letter is application layer
    iv. Multiplex and demultilplex (core process of transportation layer)
        1. Demultiplex = uncover the message process , from one branch to many
            a. process =
                i. host receive IP datagrams, host uses IP addresses (house location) and port numbers (specific kid location)
            b. Demultiplexing types
                i. Connectionless demultiplexing = specifying source and destination local port only (total 2 element)
                ii. Connection oriented demultiplexing = uses both source port and ip address (total 4 element)
        2. Multiplexing = used when sending message, handle data from multiple socket, and add header, combine data into one segment, from many branch to one
c. UDP
    i. segments may be lost

ii. upkeep rate in exchange of higher loss
iii. used in streaming apps (rate is most important)
iv. detecting error
    1. checksum = detect error by using sum data
v. uses best effort service = send and forget
vi. advantage
    1. no setup/handshake needed
    2. can function when network is compromised
vii. Spotify 2010 gunnar kreitz

LESSON 30 October 2025 – Chapter 3

Principles of reliable data transfer

a. Main activity of sending package = sender send package, receiver receive and acknowledge
    i. Possibilities = received but not acknowledge, still to be acknowledged, not yet received
b. Go back n algorithm == consisting sender window, sender, and receiver
    i. Sender
    ii. Receiver
        1. ACK = acknowledge
        2. ACK only = always send ACK for correctly received packet with highest in-order sequence
    iii. If a packet is loss, the next element will be received but will also be discarded and it will resend the latest ack before error: meanwhile sender will keep on sending pkt-n until pkt runs out and the process will be refreshed from the previously failed sent package
c. Selective repeat algorithm = will resend the package if the package failed to be sent
    i. The receiver will continue sending back acknowledge normally even when there is error previously

Connection oriented transport TCP slides

a. Characteristic of TCP
    i. Point to point (1 sender 1 receiver OR host a host b), reliable, full duplex data
    ii. Uses cumulative ACK
    iii. Is more reliable (less fast but is more accurate) than UDP (because it uses more attributes)
    iv. Uses connection oriented **(connection oriented == handshaking)**
b. TCP Segment structure: total length = 32 bits
    i. Source and dest port
    ii. Sequence number = Number of first byte in segment's data
    iii. Acknowledgement number = Sequence of next byte expected
    iv. Length of tcp header and receive number
    v. Checksum (also exist in UDP)
    vi. TCP options
    vii. Data sent by application into tcp socket
    viii. Implementation TCP == host A and host B
c. TCP timeout
    i. TCP timeout value scenarios
        1. If its too short = premature timeout, unnecessary retransmission
        2. If its too long = slow reaction to segment loss
    ii. TCP timeout formula = (1-a) * *estimatedRTT* + *a* * sample RTT
        1. a = is a constant , valued at 0.125
    iii. TimeoutInterval = EstimatedRTT + 4 * Dev RTT
        1. Dev RTT = safety margin == (1-B) * estimatedRTT + B * (sample RTT - estimated RTT) where B = 0.25
    iv. TCP Sender
        1. Sender make segment and create timer for estimating RTT
    v. TCP implementation between host a and host b
        1. Host a send seq 92+8 to host b
        2. Host b acknowledge 100
        3. Acknowledge is lost
        4. Host A retry to to send packet
        5. Successful and acknowledge also successful
        6. Scenarios
            a. IF Timeout during sending process, host A resend the packet
            b. IF packet loss, next packet wont be sent until the lost prev packet is sent
            c. The latest packet == smallest sequence
    vi. TCP fast retransmit = if a packet is lost from Host A, host A will continue on sending packets until it ran out while host B will keep on asking for the lost packet A (by sending acknowledge lossPacket), depending on timeout duration
    vii. TCP flow control diagram (in order, top to down)
        1. Application process
        2. TCP socket receiver buffers

3. TCP code
4. IP code
5. Sender

viii. Flow control = receiver controls sender with flow control to limit bytes to be sent, else there will be overflow
    1. Containing buffered data as extra remaining space

ix. TCP connection management with handshake
    1. Type1 = 2 way handshake
        a. Weaknesses = losing data, message reordering, variable delay
    2. Type2 = 3 way handshake – example in real life is human climbing where human ask server is rope attached (safe to use) and only then he will start climbing
        a. First message sent from client to server
        b. Server responds by saying its available to connect
        c. After that server is available, client send connection request
        d. Only then the server and client are connected

<u>Principles of congestion control</u>

a. Too many sources sending too much data to fast for the network to handle
b. Scenario cases – modeling interaction between router (receiver) and sender
    i. Case 1
        1. One router, infinite buffers
        2. input , output link capacity: R
        3. Two flows
        4. No retransmission needed
    ii. Case 2
        1. One router finite buffers – thus packet is dropped at router due to full buffers
        2. Idealization = when the sender only sends when it knows a router buffer is available
        3. Due to congestion, there are unneeded retransmissions resulting in data loss
    iii. Case 3 – interaction between router and hosts (sending between hosts through highway roads known as router)
        1. Consisting of four senders (host)
        2. But there are buffers in routers because same router are being used by multiple hosts
        3. End result is when throughput == 0, meaning buffer zone is full
c. Causes of congestion
    i. Throughput can never reach capacity
d. Congestion control approaches – only used in TCP
    i. End-end congestion control = taken by TCP
    ii. Network assisted congestion control = router provide direct feedback to sending/receiving hosts

iii. AIMD = sender increase sending rate (increase by 1) and decrease it (by half) when loss event happened
iv. TCP slow start = when connection begins, sender increase rate exponentially until first loss event
v. TCP cubic = instead of reducing total rate, focus on the point of congestion
1. Uses the r to the limit but not causing congestion – keeping the pipe just full but no fuller

e. Relation between RTT and throughput =
i. rtt is stable and increase once congestion is reached
ii. Throughput is increasing until it stabilizes once congestion is reached
f. Explicit congestion notification (ECN)
g. TCP fairness = when two TCP sessions share same bottleneck link, each have average rate of R/K
i. Tcp is fair only when same RTT
h. Fairness in UDP vs TCP
i. UDP doesnt use congestion control thus is not fair but its not necessary anyway
ii. TCP is fair and can do multiple packets at once

Lesson november 6

1. Transportation layer is done
2. Steps of TCP python – client side
    a. make main function
        i. main diff bween udp and tcp = tcp has connection managed by three way handshake, udp not at all
        ii. define host and port
            1. connect the socket, encode the message in bytes (messages are initially in strings)
3. Steps of TCP in C language – client side
    a. main function
        i. create buffer zone
        ii. define port and address
        iii. make listen() function
        iv. bind the listen unction to establish that the server is ready to listen
        v. accept the connection (part of the three way handshake)
        vi. open and close the socket
        vii. while still below buffer, client read the server message
4. postman server module (make coding easier such as var.get())
5. HTTP request message format
    a. Request line == method, URL, version
    b. Header lines = header field name, value
    c. body

    d. cr f = \r\n
6. POST line example =
    a. ‘POST
       Host:
       Content-type:
       Content-length
       \r\n len(body)
7. GET line example
    a. GET /products HTTP
       ‘Host: localhost:8080
       ‘\r\n
8. CONNECT function
9. Body uses json format
10. steps of POST implementation =
    a. get into terminal
    b. run virtual env
    c. run file
    d. port is now ‘listening’ to port 8000
    e. client will send the
            POST
            HOST
            Cotent Type
            Content-length

            {body json data}
    f. Server returns 201 code if succeed
11. Steps of GET implementation
    a. connect to server address
    b. make header containing
            GET
            Host:
            \r1n host port
12. GET will retrieve db content, PUT will update existing value, POST will make new value, DELETE will delete **Codes regarding the subject is in google colab notebook**

Lesson november 7 – network layer slides

Network layer overview

- IP protocol are unique (private network
- inside a port = input port, switching, output port, buffer management, scheduling
- Generalized forwarding = controlling a network from software using match+action algorithm
- datagram == the packet sent in the network layer
    - header
- Layers revisit

    - application
    - transport (has possibility of unreliability)
    - network = first layer that exist in all layer
    - link
    - physical
- network, link, and physical layer exist in all layer
- router
- network layer roles =
    - forwarding = moving packet from router input link (connect a cable) to router output link ; forwarding is located in the middle bween input and output link; example= getting through single interchange
    - routing = determine route taken by packets from source to destination; example = planning trip from source to destination

- Types of planes
    - Data plane = decide where a datagram arriving on router input port to go to which output port whereas values are indicated in packet header (within a single router)
    - Control plane = determine datagram is routed among routers along end to end path (between many routers)
        - types = traditional per router routing and software routing
        - traditional per router control plane = individual routing algorithm in each and every router in the network; using routing algorithm
            - top = control plane ; bottom = data plane
        - Software defined networking control plane
- Network layer service mode -- implement the best effort service model ---
    - because its simple and it has sufficient provisioning of bandwidth
    - it uses load balancing
    - it uses elastic congestion control

inside a router
- parts
    - input port (blue cable)
    - antennas (output port)
- architecture
    - Top = routing processor (uses firmware or microprocessor)
    - bottom = high speed switching fabric, also called as forwarding data plane (doesn't use microprocessor)
- input port elements
    - physical later
    - link layer
    - decentralized switching (where the queue and algorithm is at)
        - destination based forwarding algorithm -- computational complexity of O(n)
            - table consisting of destination address range (destination of packet) and link interface (the ID of destination)

- IF certain common numbers match, they represent an action ID (link interface)
- longest prefix matching => use longest address prefix that matches destination address -- ONLY include the common parts and censor the others; example 14**
- example of network companies in real life == cisco

- Switching fabrics = transfer packet
  - switching via memory
  - via bus => bus contention -- meaning switching speed is limited by bus bandwidth
  - via interconnection network == multistage switch -- meaning switching bween processors
  - can be scaled by using multiple switching planes in parallel (known as parallelism)
- port queueing =
  - input port queueing = happen if switch fabric is slower than buffer input
  - output port queueing => buffering = required when datagram arrive from FABRIC faster than link transmission rate, use drop policy to drop datagrams when theres no free buffers
- scheduling discipline
- deciding buffer value = RTT x C ; whereas C = capacity

- buffer management ==
  - structure => switch fabric -> datagram bugger -> link later protocol -> line termination
  - abstraction queue
  - actions of management
    - drop (might cause problems due to not aligning with TCP protocol (all packet must be complete))
    - marking = mark packet to signal congestion with congestion flag

  - Packet scheduling == first come first served, round robin, priority
    - priority scheduling = arriving traffic is classified n queued based on class; the ones with buffered packet will be prioritized
    - round robin scheduling = send data from each class in turns
      - weighted far queueing (modified round robin) = each data has weight w and gets weighted in each cycle

- Network neutrality = ISP should allocate its resource fairly by scheduling n buffer management

IP protocol
- the only layer in the network layer
- Datagram structure -- layers in order
  - version, head length, type of service, length,
  - 16 bit, flags, fragment offset
  - time to live, upper layer, header checksum
  - source IP address
  - destination IP address (32 bit)

- options
- payload data

- Main definition
  - IP address = 32 bit identifier associated w each host or router interface
  - interface = connection bween host/router n physical link, where each router have multiple interfaces, where each host have one or two interfaces

- how interfaces are connected
  - SUBNET = device interfaces that can physically reach (three host can see each other without going to an ethernet) each other without passing through a router
    - Subnet enable isolated and safer network ; example == subnet mask: /24
    - IP address structure = subnet (first 24 bit, fixed), and host (last 8 bit, changeable)
      - MAX amount of Host in a network = 256

Lesson December 5 missed lecture– network layer slides

Last lecture progress = IP protocol slides 4
Today's missed lecture = network layer, chapter 4 done completely, until Dijistra algorithm chapter 5
Target = finish slide chapter 4 network layer
terms gained

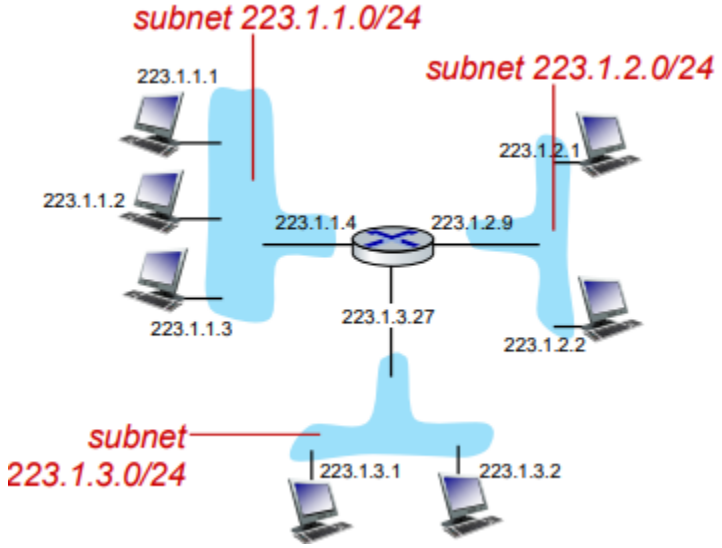
1. Host = is an object word, NOT action word

slides chapter 4 review

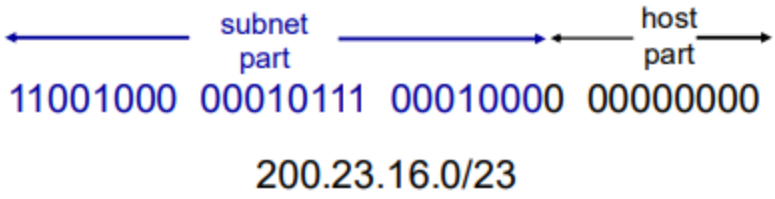
1. IP protocol continue – start pdf page 47

a. Subnet = device interfaces that can physically without passing through an intervening router

i. IP address structure = subnet (high order bits) and host part (low order bits)
ii. Subnet = isolated network of devices before passing router



iii. CIDR (IP addressing): classless interdomain routing

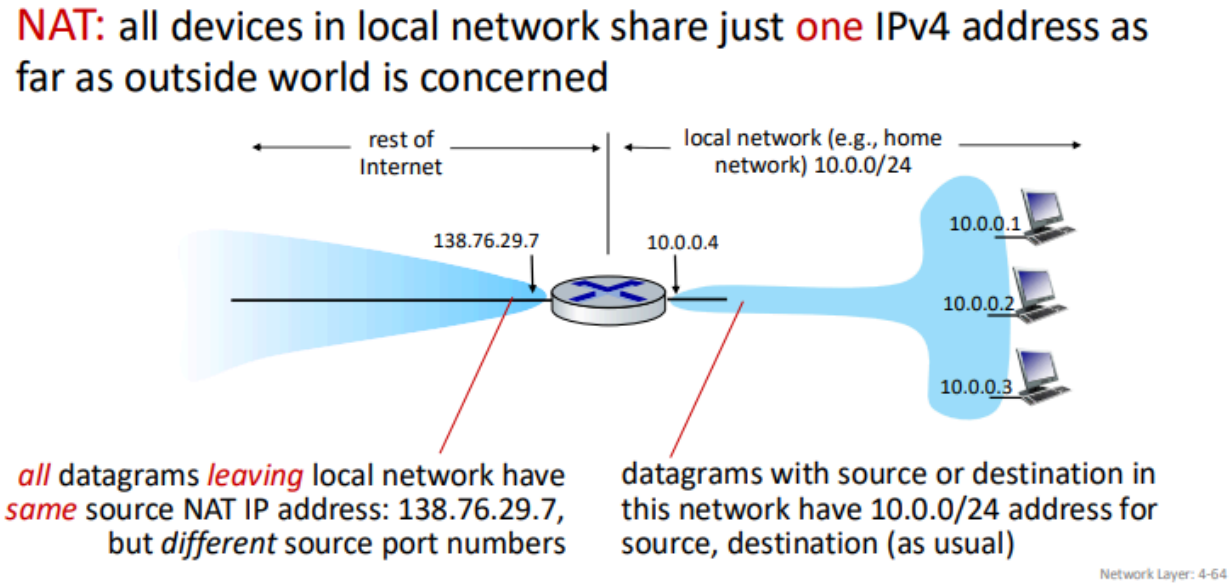


iv. DHCP (Dynamic Host Configuration Protocol) = Host obtain IP address from network server when it joins network
    1. Types = discover, offer, request, ack

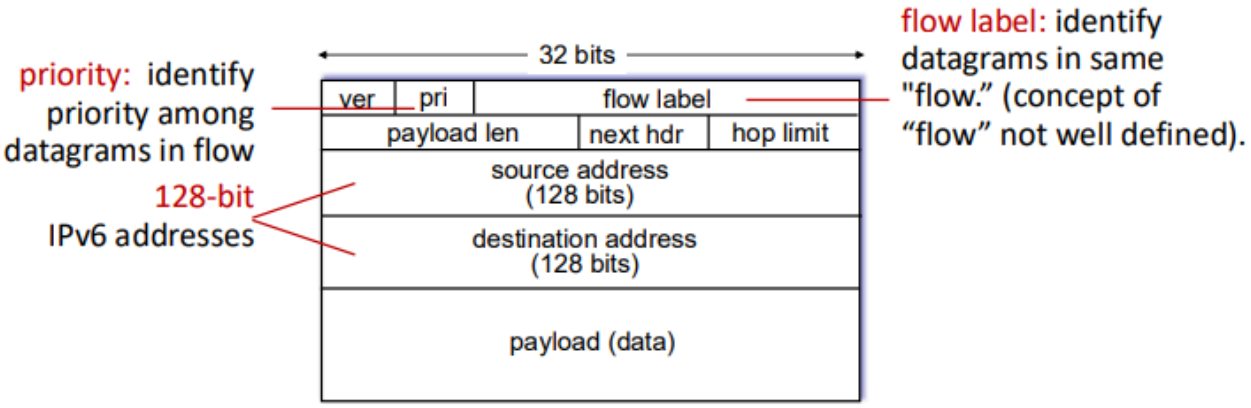
b. Getting IP address method
    i. Hierarchical addressing: route aggregation
    ii. How ISP get block of addresses = ICANN method

2. Network address translation pdf pg 64

a. NAT = all devices in local network share just one IPv4 address as far as outside world is concerned
    i. Advantage = just one IP address from provider for all devices, can change address and ISP without notifying outside world
    ii. Implementation of NAT = replace outgoing and incoming datagram's source IP address, remember every source IP address translation pair



b. IPv6 = to replace 32 bit IPv4
    i. main FORM of IPv6



What's missing (compared with IPv4):

- no checksum (to speed processing at routers)
- no fragmentation/reassembly
- no options (available as upper-layer, next-header protocol at router)

ii. Tunneling = IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers ("packet within a packet")

3. Generalized forwarding pdf pg 78

a. 'Match plus action' abstraction = match bits in arriving packet, take action
   i. Destination based = forward packet based on destination
   ii. Generalized forwarding = many header fields thus many possible action

b. Flow table abstraction
   i. flow: defined by header field values (in link-, network-, transport-layer fields)
   ii. generalized forwarding: simple packet-handling rules
   iii. Element of table = match, actions, priority, counters

c. Openflow = another form of flow table
   i. Openflow abstraction = uses match+action that unified diff kind of devices
   ii. Elements

Router
- *match:* longest destination IP prefix
- *action:* forward out a link

Switch
- *match:* destination MAC address
- *action:* forward or flood

Firewall
- *match*: IP addresses and TCP/UDP port numbers
- *action:* permit or deny

NAT
- *match:* IP address and port
- *action:* rewrite address and port

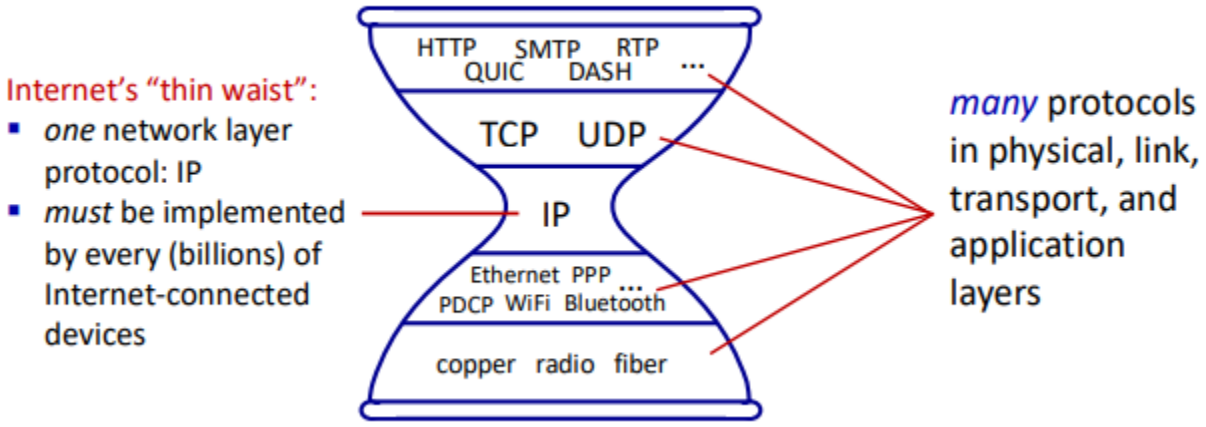
d. Generalized forwarding summary

Generalized forwarding: summary

- "match plus action" abstraction: match bits in arriving packet header(s) in any layers, take action
  - matching over many fields (link-, network-, transport-layer)
  - local actions: drop, forward, modify, or send matched packet to controller
  - "program" *network-wide* behaviors
- simple form of "network programmability"
  - programmable, per-packet "processing"
  - *historical roots:* active networking
  - *today:* more generalized programming: P4 (see p4.org).

a. Middleboxes
   i. Definition = any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host
   ii. Subcategories of middleboxes (where they are located) = NAT, Firewall IDS, load balancers, app specific, cache

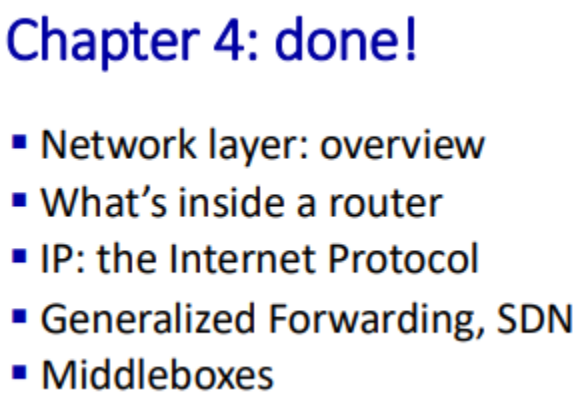
4. Internet architecture pg 92 pdf
   a. Main model =>



   b. End to end argument = network functionality to be implemented in network edge
   c. IP fragmentation/reassembly
      1. Network links have MTU (max transfer size), diff link type, diff MTU

2. Large IP diagram is divided within net where one large datagram is divided to smaller datagrams

Chapter 4 end –



Role of Control Plane = compute the forwarding tables (destination based) or flow tables (generalized forwarding)

Class 09/12/25
Progress = Chapter 5

1. Routing algorithm – chapter 5

- convert from ipv4 to ipv6 with 128 bits
- Routing algo classification
    - Global
    - Decentralized = iterative processs of computation
    - Static
    - Dynamic
- Dijkstra (centralized) link state routing algorithm
    - Centralized = all nodes have same information
    - use least cost path
    - use iterative
    - Definition = compare two alternative method of approaching a node and find the smallest/shortest path with min(path1, path2)
    - if original node and target not directly connected, path is infinite – no direct link
        - combination will be compared with direct route
    - Algo complexity = O(nlogn)
    - Message complexity = O(n^2)
    - Dijkstra route oscillation
        - when different route change costs, some part are more congested
- Bellman ford algorithm
    - D (v Z) = direct path from v to Z skipping steps
    - C (v Z) = direct path without skipping steps
    - Count the min of (C (v z) + D (v z)) for the same z with diff approach
- Distance vector algo

- Steps
    - each node waits for update or notification from neighbour
    - Recompute DV estimates from neighbours DV
    - if DV to any destination, notify neighbours
    - compare the min of three parameters
        - Example = D(ba) = min(path1, path2, path3)
        - Dba = min (C (ba) + D (aa) + ….
- Iterative communication
- Link cost changes =
    - node detects local link cost change
    - count to infinite problem
    - Message complexity
        - LS = O(n^2)
        - DV = depends on exchange bween neighbours
- Conclusion = there are two algo types that is being compared = LS and DV, LS is slightly faster

New unit in chapter 5 – Scaling Routing

- network plane controls the ISP
- autonomous systems (AS) OR domain = aggregate routers into regions
- Two types = intra AS (inside) and inter AS (outside)
    - Example of intra = RIP protocol
    - Intra uses open shortest path first (OSPF)
        - local area n backbone
- BGP protocol = messages echanged over TCP connection
    - Open,update keepalive, notification
    - AS path = list of domains
    - Next hop
- Last progress = BGP algorithm

Lecture 17 december 2025
Theme= Slides Link layer – layer 6
Chapter 1

- Nodes in link layer = host and routers
- Purpose of link layer = transferring datagram from one node to physically adjacent node over a link
- Consisted of several links
- Transportation analogy – can be private or public transportation
    - Datagram is the tourist
    - Transportation mode is the link layer protocol
    - communication link (link between two nodes or parties) is the Transport segment
    - Routing algorithm is the travel agent
- Link layer services

    - Link antennas can interfere with each other
    - Flow control
    - Error detection = signal overlap and produce noise
    - Error correction
    - Half duplex and full duplex
- Host link layer implementation
    - Use network interface card to implement link and physical layer
    - Combine hardware software and firmware
    - Each **interfaces** consist of
        - CPU
        - Controller, physical layer
- Mac address is got when bought and cant be change, while IP address is changeable and reassignable

Chapter 2 - error detection

- EDC = error detection and correction == Given two datagrams, check if they have error
- Single bit parity = detect single bit error
- Parity checking
    - If number of '1' is odd, output is 1
    - Else output is 0
    - Apply transversal to both row and column
- Checksum (appeared in transport layer)
    - Sender do addition of segment content
    - Receiver check for error
- Cyclic redundancy check (CRC)
    - Formula = d * (2^r) XOR R == nG
    - If non zero remainder == error
    - Example output ⇒ R = 0 1 1

Chapter 3 multiple access protocol

- Types of links =
    - point to point (PPP) = for dial up access
    - broadcast = radio, old school ethernet
- Collision = two or more signal at the same time
- An ideal multiple access channel
    - R = rate of sending
    - R/M = rate of sending M nodes
- Three classes of MAC
    - Random access
    - Taking turn
    - Channel partitioning
- Method MAC 1 TDMA = time division multiple access
    - Access to channel in rounds
    - Unused slots go idle
- Method 2 = FDMA
    - Uses frequency instead of time, whereas frequency = (1 / time)

- Method 3 = Random access protocols (dynamic)
    - Allows collision and recover from collision such as through delayed retransmission
    - Example = ALOHA
        - Nodes are synchronized
        - If no collision node can send in the next slot
        - If collision node retansmits frame in subsequent slots using randomized probability
            - Randomized probability will decide if a node will be sent or not
    - Only ONE node/packet can be send at once
    - Is decentralized equate no controller telling nodes to be sent in sync
- Method 4 = CSMA Carrier sense multiple access – still part of random access
    - Listen before transmit, channel will decide when to speak (transmit)
    - CSMA/CD = CSMA with collision detection
        - Reduces channel wastage, effective in wired not wireless
        - If idle == start frame transmission
        - If busy == wait till channel idle then transmit
        - Computation complexity depends on nodes
- Method 5 = taking turns Mac protocol
    - Polling = invites other nodes to transmit in turn
    - Issue in overhead and latency
    - Token passing = token acts like turns or right of speaking for each nodes
        - If a node has nothing, token will be passed, else the data owned by node will be passed around
- Cable access network = FDM, TDM, and random access
    - ISP -> CMTS (cable modem termination system) -> homes -> devices at homes
    - Cable access network
        - DOCSIS = data over cable service interface specs

Chapter 4

- MAC address == lan address == physical == ethernet address
    - Each LAN interface have 48 bit mac address
    - Analogy of MAC address = social security number
    - Analogy of IP add = postal address
    - ARP table = contact list of a node
        - If A doesnt have B contact in its ARP table, it will call everybody (broadcast address) until it founds B
        - After B receive, it will return confirmation to A, then A add B to its contact
        - Condition is, B has to be at the same router as A; A knows IP and mac address of Router and IP address of B
- **Fortinet certification program – partnership w unime**
- Ethernet = wired LAN technology – used CAT wire system
    - Ethernet frame structure == addresses => 6 byte source, destination MAC addresses
    - Properties = connectionless, unreliable

LECTURE 18 december 2025

Chapter 1

- MPLS routing = label switched router – alternative method of IP routing
    - Main feature is Flexibility = mpls forwarding decision may vary
    - alternative method of IP routing
    - Label color = violet, color of traditional IP routing = blue
- Data center
    - Involve multiplexing and hundreds if host
    - Elements
        - Server rack = contains server blades and host
        - Top of rack (TOR) switch = one rack that contain many racks
        - Tier 2 aggregation switches = connecting 16 TORs, can be in diff areas
        - Tier 1 core switches = connects 16 T2, with many connection for redundancy purpose
        - Border routers = top of the hirerachy
    - Many to many connection
        - Each specific element represents a laptop or a host, representing one ip address
        - Each application in that computer represents a port
        - Then each app do demultiplexing and etc.
    - Redundancy = when one path doesnt work, take another route to achieve the same objective
    - Load balancing (application layer) =
        - Receive external user request and direct workload within data center, returns result to client
        - Automatically choose the path
        - SDN = software defined network == so that we dont have to physically go to the racks to add a database feature
    - Environmentally healthy data center == placed in cold places like arctic to help cooling, use data center heat to heat houses

Chapter 2 - unit conclusion

- Learned stack = application, transport, network, link
- Involved groups
    - Group1 = local laptop
    - Group2 = comcast network
    - Group3 = webserver google
- Main phases
    - HTTP
    - TCP
    - IP
    - ETH
    - Physical
- Steps
    - Arrive

        - Request google
        - Go to google
        - In application layer request to DHCP
        - DHCP assigns IP address (5 methods)
        - DHCP ack the IP address
        - Cleint now has IP address, name and address of DNS
    - Client need to use google DNS before sending HTTP request
        - Client send IP datagram containing DNS query
        - Routing done by RIP, ISIS, and BGP, or OSPF,
    - Move to tcp layer, use handshaking
        - Client knows that it can connect

Chapter 3 - peer 2 peer demonstration

- Rendezvous = a table that maintain list of all the peers
    - A given node, P1 enters rendezvous, rendezvous returns list of currently available peers in table towards P1, which is none since the table is empty so far.
    - If peer want to communicate, they just need the ip address of the other
- Implementation uses python code
    - Socket can only be run once

```
PS D:\File_study_univ\semester3\ComputerNetwork> python peer.py
usage: peer.py [-h] -r PEER_READER_PORT -w PEER_WRITER_PORT -i NODE_ID
peer.py: error: the following arguments are required: -r/--reader-port, -w/--writer-port, -i/--id
PS D:\File_study_univ\semester3\ComputerNetwork> python peer.py  -r 6003 -w 6004 -i 2
binding reader peer ...
binding writer peer ...
sending peer sync ...
sending peer sync ... completed
> peer recv {             "node_id": 2,             "type": "peer_sync_reply",             "neighbours": {"2": {
: "127.0.0.1", "destination_port": 6003}}         }
> peer recv {         "node_id": 2,         "payload": "Hey there",         "type": "peer_data"     }
Hey there
```

- Code is in vscode
- Peers can be added and the message can be sent to every peers still
- Parameters
    - ‘I’ = ID of the node
    - ‘R’ = read from who
    - ‘W’ = write to ho
- Connection with OS = use mutex lock
- Threads are use for concurrent cpu use
    - Concurrent != Simultaneously

EXAM info
30 questions in 45 minutes, multiple choice, can be both theoretical (what this protocol does or delay propagation), no negative marking
Optional oral, averaged with written, three questions
Socket datagram, socket stream,
Demultiplexing: understand ip source, destination port,

Start contacting internship by october 2026
Thesis request made at least three months before graduate

- Example = november 2026 send

Exam will be on 18 January 2026