Date

→ CHAPTER #4

→ NETWORK LAYER

↓

Transport pkts from sending to recieving host

↳ On sending side encapsulate segment into pkts (include headers)

↳ On recieving side deliver segments to transport layer (remove headers)

↳ Network layer is implemented on both host and router

↳ that's why routers not see port # etc it just see IP in header field

→ Two key Network Layer Function

1) Forwarding:

↓

move pkt from router's input to router's output

2) Routing:

↓

Determine route taken by pkt from source to destination

Date

Note:

When we do routing, routing algorithm is run, as a result forwarding table is formed. Every router has its forwarding table

→ Interplay between routing and forwarding

↳ When a value arrive in pkt's header we search that value in forwarding table and see corresponding output link

∴ routing algorithm determines overall path from src to des

∴ forwarding table determines path from router to router

Date

• Internet Protocol:

> Internet Network Layer:

↳ Requires 3 things to transfer data from src to dest:

1) IP Protocol

↓

Mechanism through which we can do addressing, tells us datagram format, and how packet will be handle

2) Routing Protocol

↓

Shortest path selection
- Static Routing
- Dynamic Routing

↳ Forwarding table is formed

3- ICMP Protocol

↓

Used for error reporting and signalling

> IP datagram format

↓

Slides

Date

> Default TTL value and Hop Limit

Time to Live — routers, devices

↳ set limit on how long a packet can travel across a network

* ↳ Each time packet passes through a router TTL (or hop limit) decreases by 1

↳ If TTL reaches 0 before pkt arrival at dest, the router discard pkt & sends an error msg.

↳ Default TTL for Linux = 64
" " " Windows = 128

↳ The initial TTL value of a pkt can hint at the OS sending it

> IPv4 addressing : introduction

- IP Address :

↳ Each IPv4 address is a 32 bit number which means there are approx 4 billion possible addresses

↳ separated by octetse (.)

192.168.0.1

Range (0-255)

Date

• Interface
↳ Connection point where a device connect to network
↳ has its own IP address
↳ Routers often have multiple interfaces because they connect multiple networks together
↳ Host usually have 1 or 2 interfaces
↳ wired & wireless interface

Note: Each interface on a device has its own IP address. meaning device with multiple interfaces will have unique IP address for each interface not a single IP address for a whole device

Eg: router might have 1 IP add for connection to local home network & another for connection to ISP

→ Parts of IP Address
1) net id (Devices on same network have same net id)
2) host id

Date

> Types of IPv4 addressing schemes
- Classful IP Addressing
- Classless IP Addressing

1) Classful IP Addressing
↳ Divide IP Address into different categories or classes based on network size
↳ First few bits of IP Address determine the class
↳ which tells us how many address are available for each net

↳ Class A (Range : 0-127)		127
Class B (128 - 191)		63
Class C (192 - 223)		31
(Multicast) Class D	(224 - 239)	15
(Reserved) Class E	(240 - 255)	15

Eg: If IP is 10.1.1.1. Its a class A IP Add bcz it starts with number 0 to 127. Here 10 is network portion & 1.1.1 is represent specific device within network

Eg: IP Add = 172.16.5.4 → Class B
→ Network ID = 176.16

Eg: IP Add = 192.168.1.10 → Class C
→ Network ID = 192.168.1

Date

→ Problem with Classful Addressing

↳ Waste of Addresses

↳ Bcz devide ~~many~~ addresses by class, many addr.. go unused

2- Classless IP Addressing

↳ To solve problem of address wastage CIDR (Classless Inter Domain Routing) allows flexible division of IP Address without stricking to strict class

↳ Instead of using predefined classes CIDR use subnet mask to define size of network

↳ In CIDR notation IP add looks like 192.168.1.0/24

↓

First 24 bits are the network part and remaing for host

→ Major Difference

- Classful: A small network could be ~~forced to use~~ class B with 65,536 possible hosts even if it only need 100

- Classless: We can specify only 128 address avoid waste by preserving add for other network

Date

→ Eg of for Classless :

- CIDR block = 192.168.10.0/28
  IPv4 add bits ↑ ↓ Network
  Host bits = 32 − 28
  = 4 = 2^4 = 16
  IP Range = 192.168.10.1 to 192.168.10.14

- CIDR Block : 192.168.0.16/27
  Host : 32 − 27
  = 5 = 2^5 = 32
  Range = 32
  IP Range = 192.168.0.17 to 192.168.0.46

→ Subnet :

↳ Smaller network within a larger network

↳ Divide larger network into smaller

↳ Divices in same subnet can communicate with each other directly without needing to go through a router

→ IP Address has 2 parts

- Subnet Part → identify specific subnet
- Host Part → identify host within that subnet

Subnet1:

Date

Eg: 223.1.1.0/24

- this subnet includes all IP address from 223.1.1.1 to 223.1.1.254
- leave 0 for subnet address & 255 for broadcast

- **Subnet Mask:**
  - ↳ 32 bit number used in IP ~~addressing~~ networking to divide an IP address into 2 parts:
    1- Network part
    2- Host part
  - ↳ 1 bit is dedicated to network
  - ↳ 0 bit to host

Eg: 255.0.0.0 in binary is
11111111.00000000.00000000.---
Network bits: 8
Host bits: 24 (32-8)

- **Subnetting a Network**

Given, IP address = 128.125.0.0
its a class B IP address
Task: Divide this IP Address into 8 subnets

Date

→ We will need to adjust the class B default subnet mask i.e 255.255.0.0

→ Goal: To create more network identifiers by taking some of host bits eg using them as part of network portion

Why take Host bits?

- In subnetting extra network identifiers are created by borrowing bits from host part
- Bits taken from host part doubles the number of subnets we can create

→ Finding how many bits to borrow?

- Number of subnets = 2^n

$8 = 2^3$

So we need 3 bits. Meaning we will add 3 more bits to the network portion of subnet mask.

→ 255.255.0.0
  ↳ 16 bits net
  ↳ 16 bits host

→ Now:
  ↳ 19 bits net
  ↳ 13 bits host

128.125.SSS00000.00

SSS	
000	Put these here one by one and do
001	
011	
100	
101	
110	8 subnets
111	

IP address available for each subnet
$= 2^{13} - 2$ (exclude 0's 1's)

Date

Binary:

11111111.11111111.11100000.00000000

Decimal: (Subnet Mask)(First Subnet)

255.255.224.0 (Internally)

New subnet mask allows network to be split into 8 subnets.
(Always same subnet mask - FLSM)

Note: All devices within same subnet have same subnet mask. Allows them to recognize they belong to the same subnet

- When packet arrives at router, which subnet to send packet to?

↳ Router do:

1- Logical AND Operation :: Takes des IP Address from incoming pkt & perform logical AND operation with each subnet mask in its fowarding table

2- Matching Subnet ID : Result of AND operation is the subnet ID. Tells router which subnet Dest IP belongs to

Date

3-Forwarding : After identifying sends

⇒ EXAMPLE FOR CLASSLESS IP ADDRE--

(1) Given, ISP's block = 200.23.16.0/20

Here, Net bits = 20

Host bits = 12

↳ meaning

$2^{12} = 4096$ IP addresses in ISP block

Task: ISP has 8 customers each requiring a block of size 512

We know, $2^9 = 512$

9 = host required

32 - 9 = 23 bits for net

Hence,

ISP devides main block (200.23.16.0/20) into smaller subnet of 512 addresses each using /23 for each subnet

Date

(2) IP's block = 17.12.14.0/26 ∴
32 - 26 = 6 host
∴ 2^6 = 64 IP addresses
Task: Organization has 3 departments each needing a different number of IPs
i) Subnet 1:
IP address require = 32
$2^5 = 32$
So organization uses
32 - 5 = 27 i.e 127 for this subnet (Range: 17.12.14.0 to 17.12.14.31)
ii) Subnet 2:
IP address require = 16
$2^4 = 16$
So 128 for this subnet
Range: 17.12.14.32 to 17.12.14.47
(iii) Subnet 3:
Require = 16 IP add
Use 128
Range: 17.12.14.48 to 17.12.14.63

Date

∴ How organization gets IP or network address that is classful or CIDR

∴ How local administration has fixed or variable length that IP address is FLSM or VLSM

∴ Classful or classless can both be subnet as FLSM.
However for VLSM it should be classless

Date

→ DHCP: Dynamic Host Config Protocol

↳ When a device like laptop connects to network like WiFi it needs an IP add to communicate with other devices. DHCP is the system that automatically provides these IP addresses.

↳ Steps of how it works:

1- Host Request an IP Address
   ↳ broadcast DHCP disover msg
2- DHCP server responds
   ↳ with DHCP offer msg
3- Host request offered IP
   ↳ DHCP request msg
4- DHCP server confirms
   ↳ DHCP ack msg

[Optional] if client remembers & wishes to use previous allocated network add

↳ If you leave network DHCP server can give that IP address to someone else

→ Network & broadcast address not assigned (first add) LO) (last add)..

∴ DHCP works at the App Layer and sends its messages using UDP at Transport Layer.

Date

∴ DHCP server is built in the router (main device that connects all smaller network or subnets)
As router has built in DHCP server it automatically assign IP addresses to all the device on each subnet it connects to
∴ Port 67 & 68 are standard port in DHCP protocol for DHCP server and DHCP client respectively.

→ Other then IP addresses DHCP can provide:
> Addresses of first-Hop-Router
↓
Device that connects network to internet or other network
↳ tells router's IP address, so that device know where to send data if reach outside local network
> DNS server address
> Network Mask
↓
to identify which part for network & which for host

Route Aggregation

Date _____

- **Root Summarization / Address Aggregation**

Org 1 200.23.16.0/23 → 200.23.00010000.0

Org 2 200.23.18.0/23 → 200.23.00010010.0

Org 3 200.23.200/23 → 200.23.00010100.0

⋮ ⋮ ⋮

Org 7 200.23.30.0/23 → 200.23.00011111.0

- Select common part

  200.23.00010000.0

  200.23.16.0/20

  ~~200.23~~

↳ By doing this we are advertising multiple address within that address

Date

- **IPv4 fragmentation, reassembly**
  - ↳ Processes used to handle large data packets over a network, especially when these packets encounter net.. with smaller maximum frame size known as **MTU**
    - ↓
    - maximum size that a single data pkt can be on a network link
    - Eg: Ethernet network might allow a pkt upto 1500 bytes
  - ↳ **Why** Fragmentation is needed?
    - ↳ When large IP pkt is too big to fit into MTU of a link on the route to its dest, it has to be broken into smaller pkts called fragments
  - ↳ **How** Fragmentation works
    1. Breaking down the pkt
    2. Header Info
       - ↳ Each fragment keep copy of original IP header & which pkt they belong to

Date

3- Independant travel
↳ Router treat each fragment like an independant pkt so they may take diff path to dest

↳ Reassambly of pkts:
↳ When all fragments arrive at dest they are reassembled back into orignal IP packet
↳ If one pkt is lost entire pkt will be resent
↳ fragments reordered after arriving out of order using header

Date

→ IPv4 fragmentation, reassemble

EXAMPLE

Total Datagram size: 4000 bytes

means Header Byte = 20

Playload Data = 3980

It will be divided into 3 ~~segments~~ fragments (MTU Limit = 1500)

F1:

length = 1500

Header = 20 PD = 1480

Offset = 0/8 = 0

F2:

length = 1500

Header = 20 PD = 1480

Offset = $\frac{1480}{8}$ = 185 ← F2 PD

F3:

length = 1040

Header = 20 PD = 1020

Offset = $\frac{F1(PD)+F2(PD)}{8} = \frac{1480+1480}{8} = 370$

Date

∴ Even though TCP tries to set the right pkt size to avoid fragmentation. IP at Layer 3 (NL) is there as a 'safety net'. If unexpected issues occur in path IP frag. breaks pkt down to ensure they reach dest.

- **NAT**: Network Address Translation
  - ↳ all devices in local network just share one (public) IPv4 address for outside world
  - → all pkts leaving local network have same src NAT IP add but diff port#

Date

• IPv6 : Motivation

IPv4 Problem 1: Limited address in IPv4

IPv4 Problem 2: Complexity & Speed in Routing
↳ have variable length header
↳ Solution: IPv6 uses fixed length header of 40 bytes

↳ IPv6 ~~allows for~~ can handle diff type of data (video, voice..)

> Transitioning from IPv4 to IPv6
↳ It is a complex process bcz not all devices can be upgraded at the same time
↳ Challenge in transition occur as there is 'No Flag Day'

↓

> Date type when every switches to new system. In reality ~~network cannot switch all at~~ once bcz diff devices & router may not support IPv6

↳ Therefore, network must operate in a mixed environ.. where both IPv4 & IPv6 coexist

Date

> How Mixed IPv4 & IPv6 Network Operate

1- Tunneling (Smuggling)

↳ allows an IPv6 pkt to be sent within an IPV4 pkt. analogue to putting a letter inside envelop (IPv4)

↳ How it works?

- IPv6 datagram is encapsulated inside an IPv4 datagram if
- IPv4 router handles pkt as it were just another IPv4 pkt, even though it contains IPv6

• Middlebox

↳ Routers usually just foward data pkts from one device to another based on des IP add but middlebox do more then that. (Inspect, alter, secure or manage data)

Date

→ Types of Middlebox

1- Firewall
2- NAT ✓
3- Load Balance
4- Intrusion Detection System (IDS)
5- Proxy Sever
6- WAN Optimizer.

Notes:

1- First 4 bits = Version of IP
Next 4 bits x4 = Header Length of IPv4
40 = Header length of ipv6.

2- Subnet mask: (How To find for CIDR)
129 means 29 bits reseved for ne.
so write 1st 29 bits ones &
last 3 bits 0
255.255.255.248

3- To find network address we do bit wise AND of b/w IP add & subnet mask