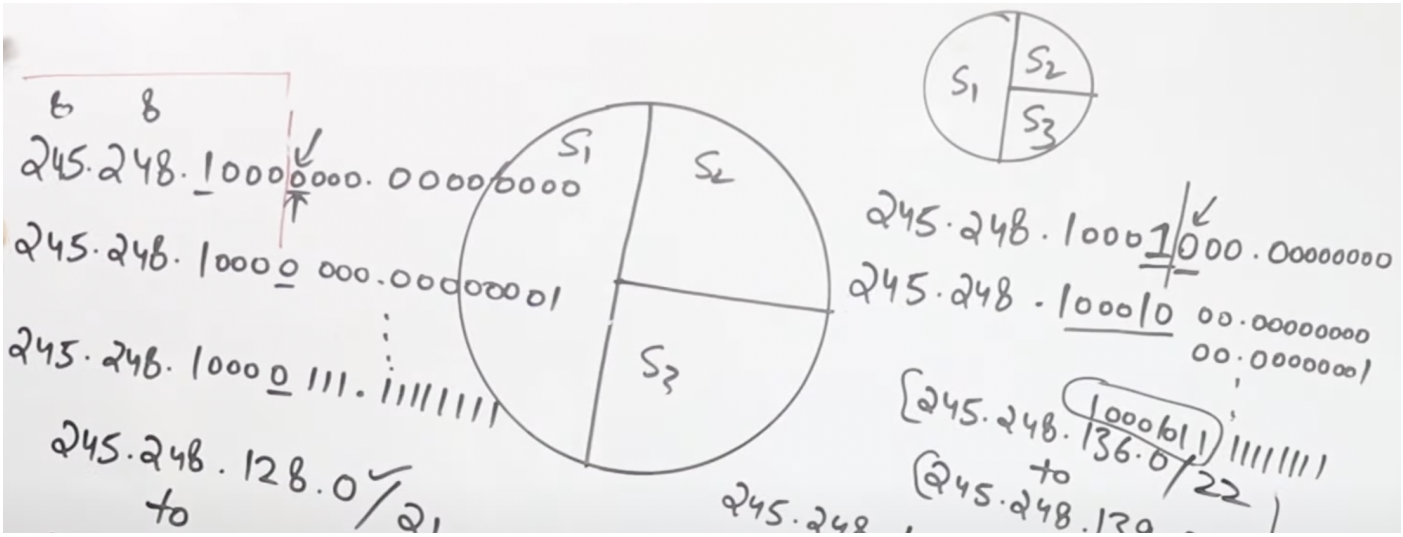
Computer Networks 2

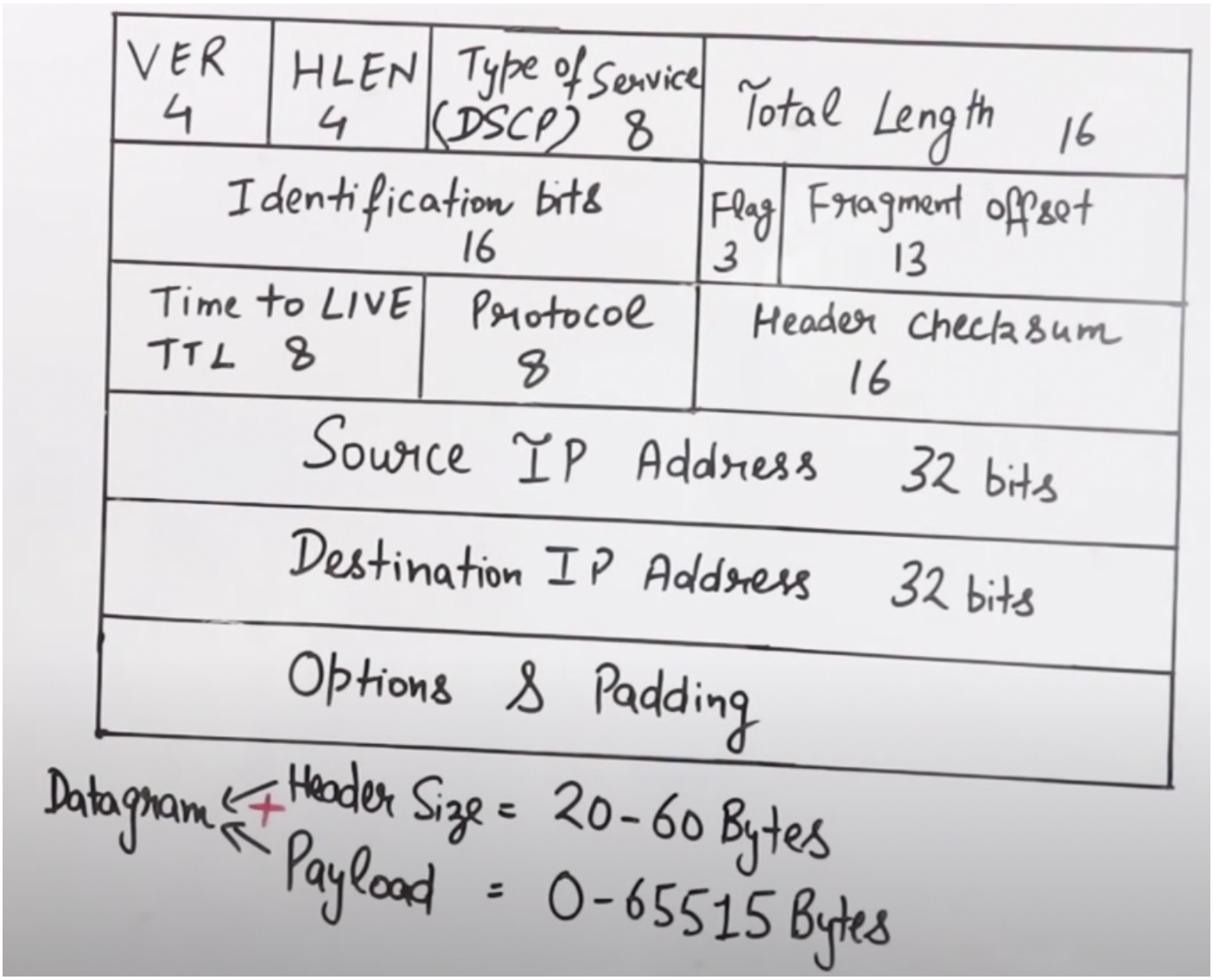
VLSM in CIDR:



IPV4 Header:

- Connectionless protocol
- Datagram service
- VER= version (IPv4 or IPv6)
  - 4 bits
  - In case of IPv4 its value is 0100
- HLEN=header length
  - 4 bits
  - size of header has to be within 20-60 bytes
  - whatever the 4 bits have, multiply it by 4
  - Example if HLEN=1010 which is 10 in decimal, then header size is 10X4 = 40 bytes
- Types of service/DSCP (Differentiated Services Code Point):
  - 8 bits
  - PPPDTRC0
  - The three Ps are for if we want to send some info about priority
  - D is for delay (1 if there should be no delay, 0 if delay is ok)
  - T is throughput (1 if its a priority, 0 if its not)
  - R is reliability (1 if its a priority, 0 if its not)
  - C is cost (1 if its a priority, 0 if its not)
  - 0 is just an extra bit for future purposes
- Identification bits, Flag and Fragment offset used for fragmentation
- Time to Live (TTL)
  - 8 bit

    - 8 bits
    - time left if stuck in a loop
- Protocol
    - 8 bits
    - any protocol that came from above layers
- Header Checksum
    - 16 bits
    - calculates checksum of values of header at each step to make sure header hasn't changed
- Options and Padding
    - Record route
        - On every hop, the new IP address is recorded
        - max 9 addresses can be recorded
    - Source Routing
        - Strict Source Routing: Decide complete route before hand
        - Loose Source Routing: Decide route partially before hand
    - Padding
        - To make header multiple of 4, add extra padding



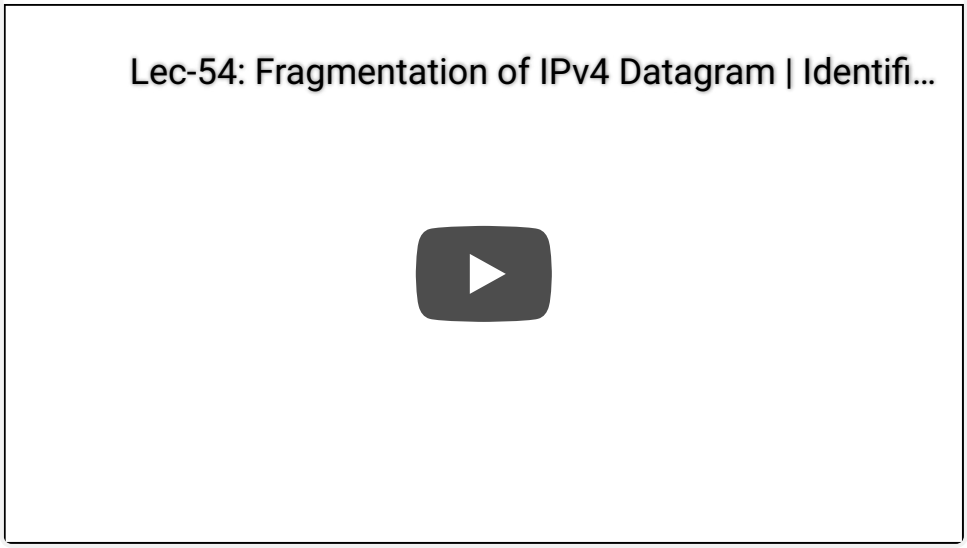
Fragmentation of IPv4 Datagram:

- Three parts:

- Identification bits (16 bits)
- Flag: (3 bits)
  - Res (reserved)
  - DF (do not fragment)
    - if fragmentation is allowed => 0
    - if fragmentation is not allowed => 1
  - MF (more fragment)
    - if no fragments present after the current fragment => 0
    - if fragments present after the current fragment => 1
- Fragment offset (13 bits)
  - No of data bytes ahead of you
  - Divide number of data bytes by 8 and then write answer
  - Example if 480 bytes (don't count header data) are before the current data, then fragment offset is 480/8 = 60

Important numerical!!

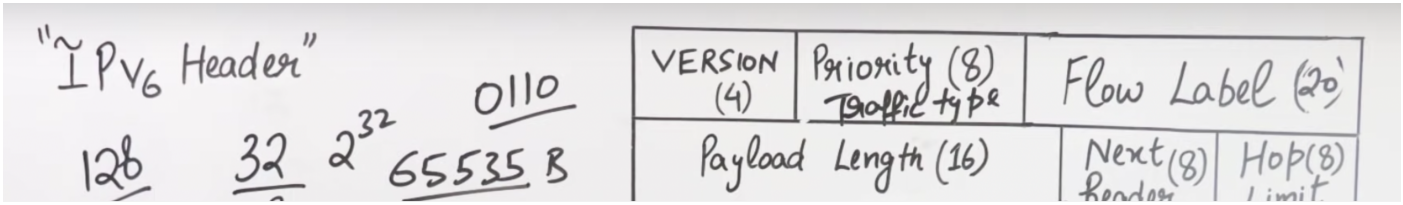
YouTube Video



IPv6 Header:

- 2^{128} addresses
- version=0110=6 (4 bits)
- Priority/Traffic byte (8 bits)
- Flow Label (16 bits): Converts datagram to virtual circuit. So if there are bits in flow label, then we reserve a path before sending. Used for real-time stuff
- Payload length (16 bits): Can send up to 4 GB data with extension headers
- Next header (8 bits): Has the identification number of extension headers
- Hop Limit (8 bits): Same as TTL(Time to live), as soon as the value is zero, the packet is dropped to prevent loops
- Extension headers:
  - Routing header(43): Decide from which which router the packet should go through
  - Hop by Hop option(0): Provide some info to each node you hope on
  - Fragment header(44): If source is doing fragmentation, add all its header here.
  - Authentication header(51): To send user password, maintain data integrity, checksums

- Destination options(60): Header that can only be opened by destination
- Encapsulating Security Payload(50): details of encryption decryption



Routing Protocols:

Routing protocols are a set of instructions by which routing information is exchanged between routers so that routing decisions can be made.

- The internet is divided into many Autonomous Systems.
- An Autonomous System (AS) is a set of Internet routable IP prefixes belonging to a network or a collection of networks that are all managed, controlled and supervised by a single entity or organisation.
- Intra domain routing protocols are used inside one AS
- Inter domain routing protocols are used between two AS
- Properties desirable by routing protocols:
  - Correctness
  - Simplicity
  - Robustness (strong to handle every situation)
  - Stability
  - Fairness and efficiency
- Two types:
  - Non-adaptive/Static Routing: Don't base their routing decisions on any measurements or estimates of the current topology and traffic. The choice of route to use to get from I to J is computed in advance, offloaded and downloaded to the routers when network is booted.
  - Adaptive/Dynamic Routing: Change their routing decisions to reflect changes in the topology, and traffic as well.
- Routing Protocols:
  - Optimality principle
  - Flooding
  - Shortest path algorithm

- Distance vector routing
- Link state routing
- Routing in ad hoc networks

The Optimality Principle:

- If router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.

Flooding:

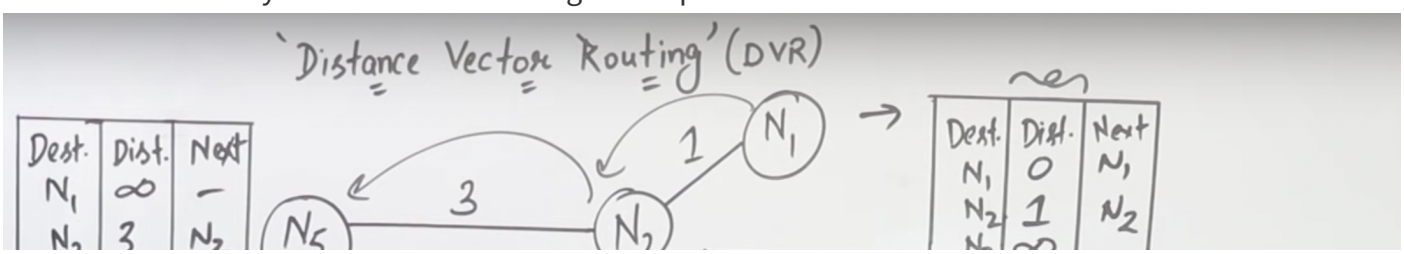
- Each router must make decisions based on local knowledge, not the complete picture of network
- Incoming packet is sent to all the neighbours and then the neighbours send it further to their neighbours until hop count becomes zero and the packet is removed from the network
- Can produce an exponential number of duplicate packets.
- Ensures that a packet is delivered to every node in the network
- Wasteful if package is to be sent to only one node
- Easy setup
- Extremely robust (will always find a way to send to all nodes)
- Very less delay
- High overhead

Shortest Path Algorithm:

- All nodes know complete picture of the network
- Build a graph of the network
  - Each Vertex representing a router
  - Each Edge representing a link
- Path length is number of hops/geographical distance etc.
- Dijkstra Algorithm used to find shortest path
-

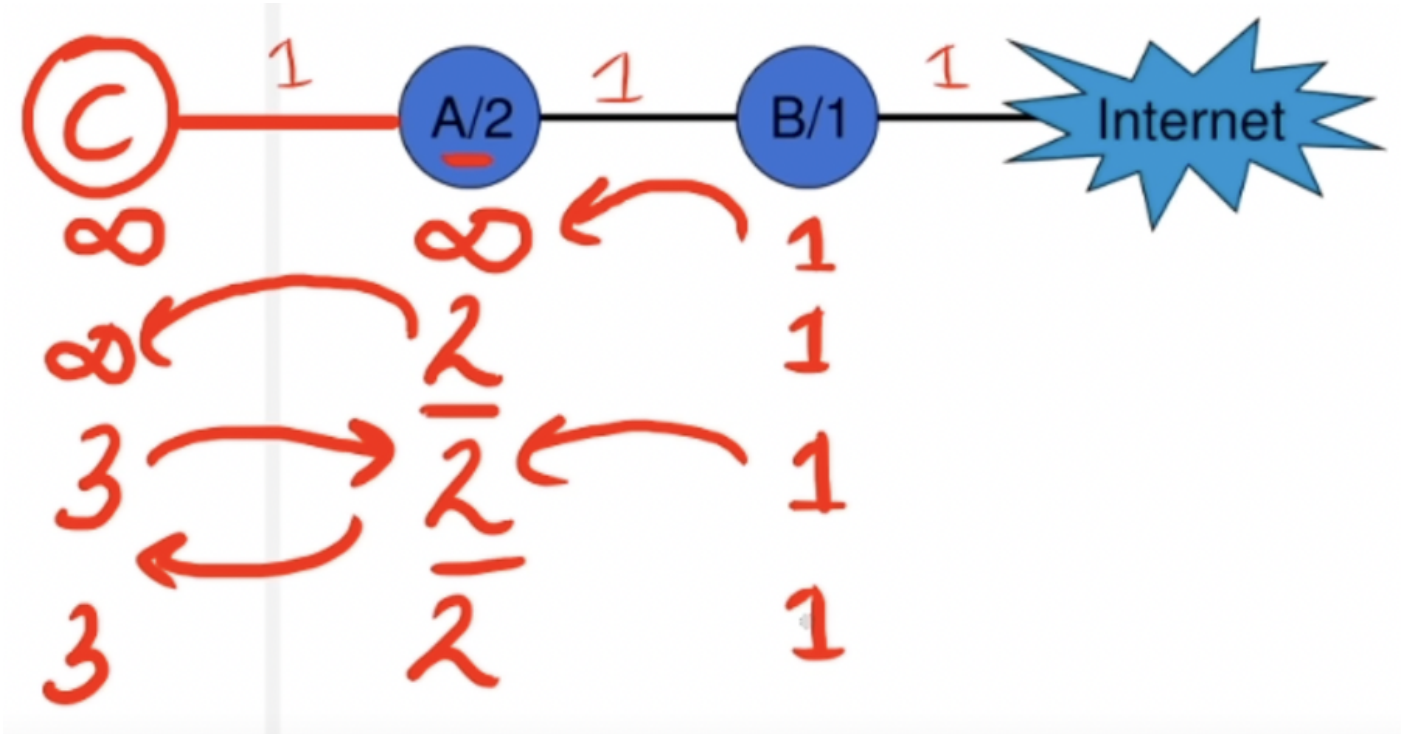
Distance Vector Routing (DVR):

- Initially all routing tables are empty
- In the first pass, each node gets the distance of its neighbours and adds it to their personal routing table
- In the second pass, each nodes shares its distance vector with its neighbours, who copy it into their routing tables.
- Many passes like this happen to make sure info reaches everyone
- Suppose you have to find N5 to N1 and you have N2's routing table. SO you find N5->N2 + N2->N1 and write the total in N5->N1
- Next is where you reach after covering the respective distance



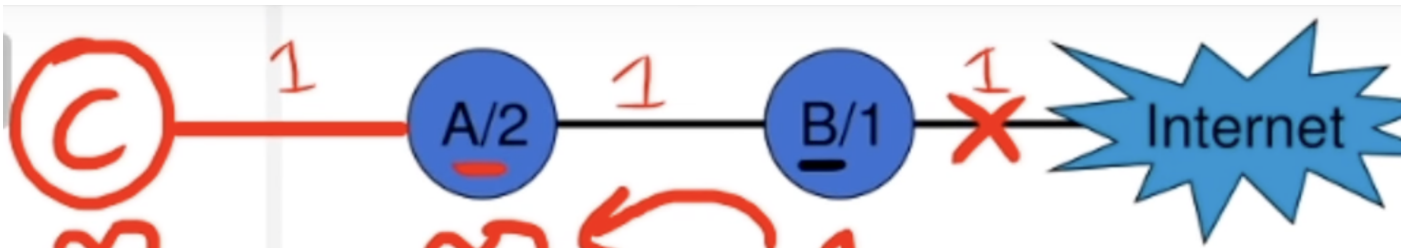
Count to infinity problem in DVR:

Normal case:



Special Case: Suppose the link between Internet and B is broken

This keeps going on and on leading to infinity



Link State Routing:

- Steps:
  1. Create personal link state tables/packets with neighbours
  2. Flooding: Everyone sends their personal tables to everyone
  3. Dijkstra Algorithm used to find shortest path
  4. New complete routing table created
- The personal link state tables contain sequence numbers, TTL and neighbours with their distance
- Link state tables:

- Applying Dijkstra algorithm at R1

- R1 final routing table

- Router keeps check with the help of sequence number if packet is new or duplicate
- If a router ever crashes it will loose track of its sequence number. If starts again at 0, the next packet it sends will be rejected as a duplicate.
- If sequence number received is 65,540 (ever corrupted) instead 4, then all the packets 5 through 65,540 will be rejected.
- Solutions to all these problems is Age/Hop count when it hits zero packet is discarded
- To improve this method:
  - When a link state packet comes in to a router for flooding, not transmitted immediately, but put it in the holding area to see if another link state packet coming from the same source router
  - If seq#s are equal, the duplicate is discarded
  - If seq#s are difference, older one is thrown out
- Packet buffer for router B:

Hierarchical Routing:

At some point it is no longer feasible for every router to have an entry for every other router, so we do hierarchical routing:

- Routers are divided into Regions. Each router knows all details within its own region but knows nothing about internal structure of other regions

Congestion Control:

- Too many packets arriving in the input line and all need the same output line
- Network and Transport layer both responsible to handle congestion
- Timed out by the time the packets get to the front of the queue. Retransmissions increase the congestion collapse.

Congestion Control Algorithms (slower to faster):

- Network provisioning
- Traffic-aware routing
- Admission controlling
- Traffic throttling
- Load shedding

Network Provisioning:

- Increase resources which can be dynamically added.
- Examples:
  - Turning on spare routes
  - Purchasing bandwidth on the open market
  - Backups lines used

- Backups lines used
- Routers and links that are regularly heavily utilized are upgraded a the earliest opportunity.

Traffic-aware routing:

- Change traffic patterns during the day as network users wake and sleep in different time zones.
- Shift traffic away from heavily used paths.
- Add Multi-path routing
- Shift traffic across routes slowly enough that it is able to converge

Admission control:

- Decrease the load
- In virtual-circuit networks, new connections can be refused if they would cause the network to become congested.

Traffic throttling:

- Request the sources to throttle or slow down their traffic
- Identify the onset of congestion by monitoring average load, queuing delay, packet loss.
- Routers must participate in a feedback loop with the sources. to tell them network in congested
- Router will find queuing delay, d. Whenever d goes above threshold, router gets to know there is congestion

  $$d_{new} = \alpha\, d_{old} + (1- \alpha)\, s$$

  where α is a constant that determines how fast the router forgets recent history and s is Instantaneous queue length
- Choke Packets:
  - Sent to notify a sender of congestion
  - Router selects a congested packet and sends a choke packet back to the source host with header bit for congestion turned on
  - Source reduce the traffic to the destination by 50% after receiving a choke packet.
  - Earlier called SOURCE-QUENCH message and now called Explicit Congestion Notification.
  - In explicit congestion notification, instead of sending a packet, they just tag a packet by setting the header bit and let it go to the destination. Destination then sends an ack message
- Hop-by-Hop Backpressure:
  - Alternative of choke packets
  - Choke packet only affects the source but here decrease the congestion on each hop as you move to the source

Load Shedding:

- When all else fails, the network is forced to discard packets that it cannot deliver.
- A good policy for choosing which packets to discard can help to prevent congestion collapse.
- Random Early Detection:
  - Running average of their queue length, if exceeds a threshold, small fraction of packets are dropped at random
  - It improves performance compared to routers that drop packets only when their buffers full.

Quality of Service:

- Application requirements: Requirements for bandwidth, delay, jitter(variation in delay or packet arrival time) and loss
- Traffic shaping:
- Packet scheduling
- Admission control
- Integrated services
- Differentiated services

ARP (Address Resolution Protocol):

- IP Address -> Logical
- MAC Address -> Physical
- If A wants to send message to C, now it knows the IP of C but not the MAC address.
- To get the MAC address, it sends its MAC address, IP address, C's IP address and in Destination MAC Address fill FFFFFFFFFFFF to give broadcast message to everyone in the network
- Then C checks that its IP address has been used so it replies back in unicast way to A with its MAC address. Now A can easily communicate with C
- Now if A wants to talk to Z:
  - A gives IP address for the default gateway/router in network 1 and broadcasts
  - Then router 1 sends the same broadcast to router 2 to get its MAC address
  - Then router 2 sends broadcasts inside its network to get MAC address of Z

- Headers:
  - Hardware Type: If we use ethernet we use 1
  - Protocol Type: If IPv4 then $(0800)_{16}$
  - Hardware Length: MAC Address length (6 bytes)
  - Protocol Length: If IPv4 then 4
  - Operations:
    - Request:1
    - Reply: 0
  - Sender Hardware Address: 6 Byte MAC Address
  - Sender Protocol Address: 4 Byte IP Address
  - Target Hardware Address: Put FFFFF... or 00000.. if you want to broadcast or send to particular system
  - Target Protocol Address: 4 Byte IP Address of destination

Network Address Translation:

- Translates private IP to public IP and vice versa
- Used in IPv4
- Range of Private IPs:
  - 10.0.0.0 to 10.255.255.255

    - 10.0.0.0 to 10.255.255.255
    - 172.16.0.0 to 172.31.255.255
    - 192.168.0.0 to 192.168.255.255
- Now the Router/NAT Device will create a translation table
- Let IP of our network is 125.12.31.7.
- But the default gateway inside our network creates private IPs for all the hosts with the IP 10.10.0.0
- Whenever a private IP(say 10.10.0.2) wants to send a message to an outside public IP(say 25.25.25.10), a record will be added to the translation table with the private and public IP.
- When the message comes back it will match the public IP of the message to the public IP in the table and send it further to the matching private IP

Transport Layer:

Responsibilities:

- End to end delivery (Port to port)
- Reliability (In-order delivery and no loss of data) - Use TCP if reliability needed else UDP
- Error Control (Checksum)
- Congestion Control
- Flow control (Stop and Wait, GO-BACK-N,Selective Repeat)
- Segmentation (Creates segments and sends it to network layer)
- Multiplexing/Demultiplexing

TCP(Transmission Control Protocol):

- Byte streaming: The continuous data flow from application layer is converted to segments/collection of bytes
- Connection-oriented: Uses 3-way handshaking protocol to establish a connection which ensures reliability
- Full duplex: When a connection is created between A and B, both A and B can send data to each other at the same time.
- Piggybacking: Ack sent with data (uses go-back-n or selective repeat)
- Error control: Checksum
- Flow control: Keep a check on how much data you're sending, it should be less than or equal to the size of the buffer of receiver
- Congestion control: Keeps a check whether the network and the receiver has the capacity of handling the amount of data going

TCP Header:

- Min 20 bytes, max 60 bytes
- Source port and destination port are 16 bit having the address of the ports
    - Total ports can be 2^{16}=0-65535
    - 0-1023 are well known ports (Important standard ports fixed like HTTP uses 25)
    - Rest all ports are either registered or non-registered ports
- Sequence number: Sequence number in each segment. Each byte in a segment has a unique

- Sequence number: Sequence number in each segment. Each byte in a segment has a unique sequence number
- Acknowledge number: Receiver sends the next expected sequence number. So if sender sends byte number 1018, ack number will be after accepting the byte 1019
- HLEN (Header Length):

Lec 65