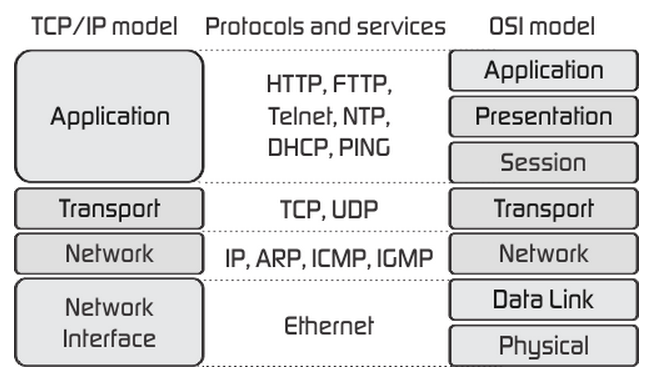
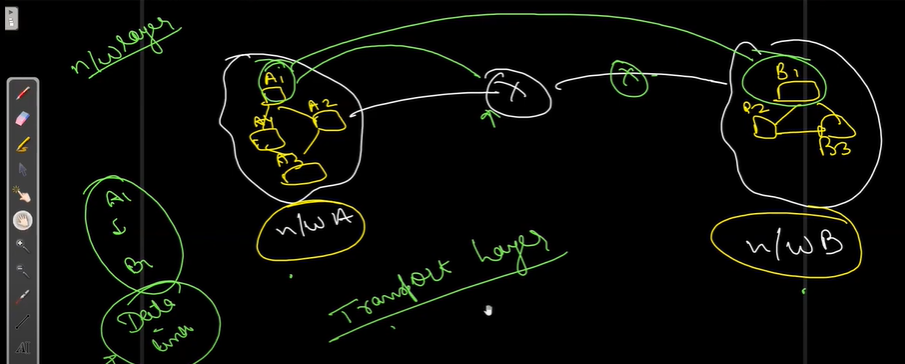
Transport Layer(TCP/IP model)



**Roles of Transport Layer**

1. [process](javascript:;) [to process delivery](javascript:;)/[end to end delivery](javascript:;)/ [port to port delivery](javascript:;).



[In this network we have different](javascript:;) [hosts a1 host](javascript:;), [a2 host, a3 host, a4 host in network A and ne](javascript:;)twork B with b1[host](javascript:;), [b2 host and b3 host](javascript:;) [we want to transfer some data from](javascript:;)  
[this network A to B from](javascript:;) [a1](javascript:;) [host to b1](javascript:;). [So,](javascript:;)

[Network layer provides host to host](javascript:;) [delivery right if we want to send a data](javascript:;) [from a1 to b1 then it connects a1](javascript:;) [via b1](javascript:;) host to host.

Data link layer takes care of where the next hop is i.e. hop to hop delivery, router to next router.

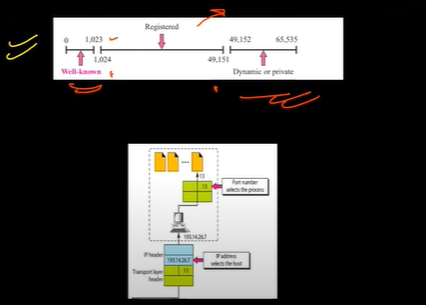
Now, Data from a1 [already reaches to b1 with the help of](javascript:;) network layer, what does the Transport layer do???



[a1 and b1 is your two systems, a1](javascript:;) [wants to send some data to](javascript:;) [b1 laptop in these laptops you](javascript:;) [have different applications open](javascript:;) [so from an application in a1 you want to](javascript:;) [send the data to the application into b1](javascript:;)

[we have identified i have to send](javascript:;) [a data from a1 to b1 but in a1 we have](javascript:;) [selected a process/e](javascript:;) or [application and from that application i](javascript:;) [want to send data in particular](javascript:;) [application of b1](javascript:;). This is taken care by the transport layer.

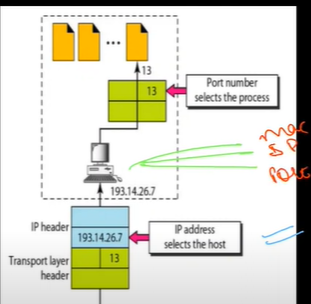
1. Addressing:-



Well known port number :- registered assigned and controlled by the icann organization

Registered port number:- neither assigned nor contolled by icann only registered to prevent duplication.

Dynamic port number:- temporary/private port number



Every computer has three addresses Mac address[physical address], i/p address[logical or network address], port address[port number]

Port number is of 16bits. Range os values it can have 0-2^16-1 = 0 –( 65,536-1) = 0 – 65,535.

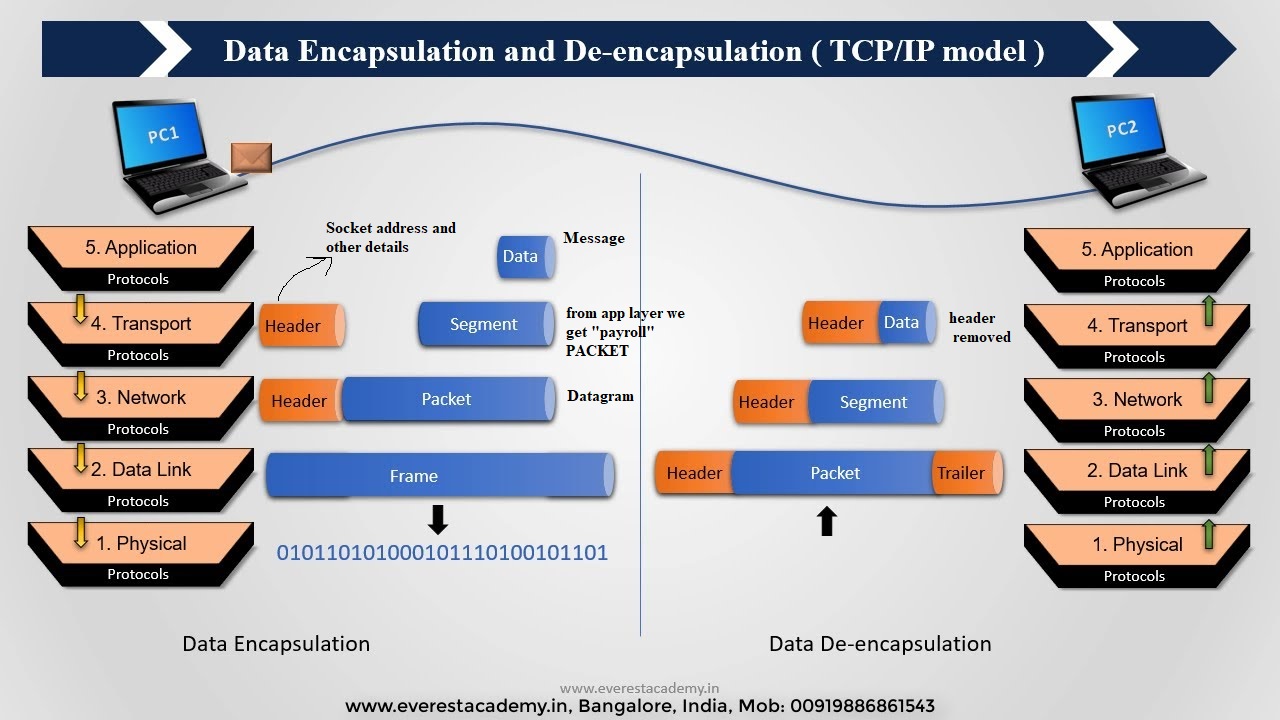
Transport layer uses the combination of i/p address[to identify the host system] and port number[to identify the process i.e. address of a particular process] to transfer data[communicate] from one process to another process.

And combination of ip/address + port number is called the **socket address.** And socket address helps to achieve process to process communication.

Building[ip address] -> Floor -> flat number[port number]

1. Encapsulation & decapsulation:-

Encapsulation:- It happens at source host. It is a process by which diff. layers of tcp/ip models adds an extra information to the message it receives or add headers to it.

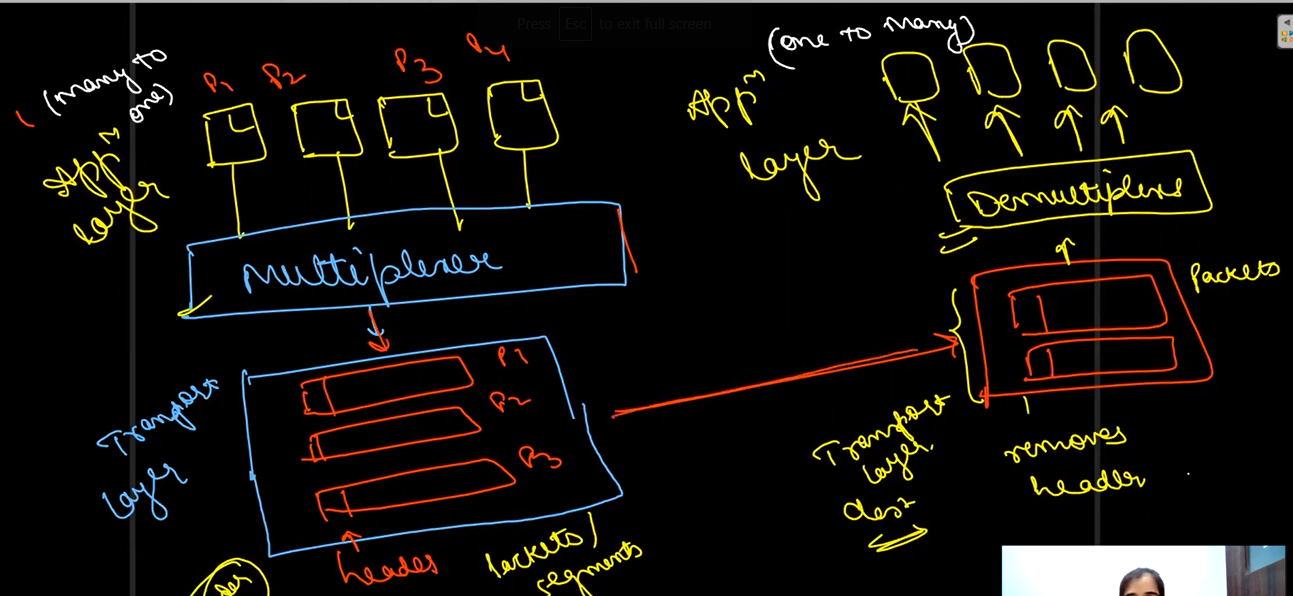


and decapsulation happens at the destination host.

1. Multiplexing / deMultiplexing:-

Multiplexing – when you receive msg from more than one source. [Transport layer at source acts]

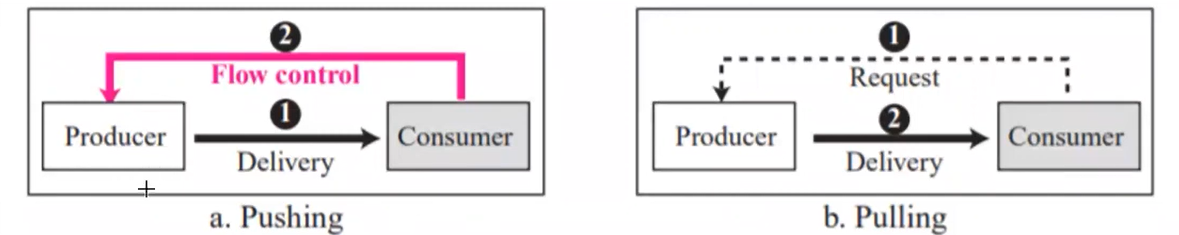
DeMultiplexing – when you send msg to more than one source. [Transport layer at destination]

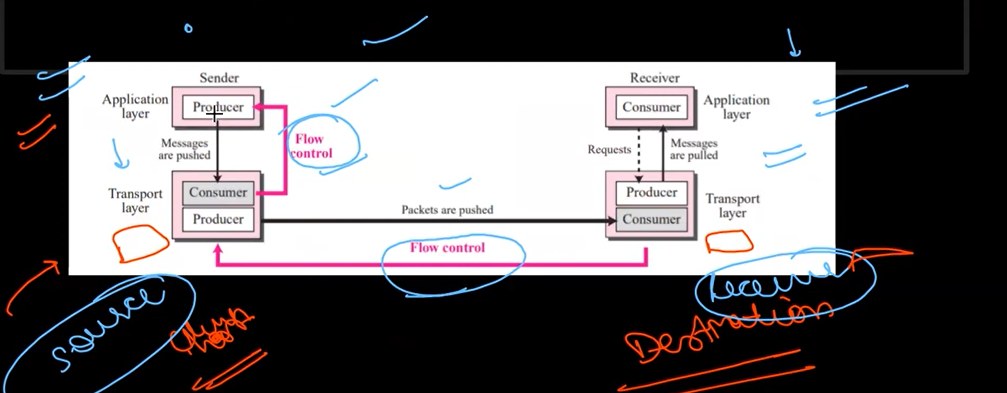


Sender

1. Important controls:- Flow control, error control, congestion control

Flow control – pushing – without asking just sending the data[flow control needed],, pulling – sending data when requesting.





Pink line is where flow control needed.

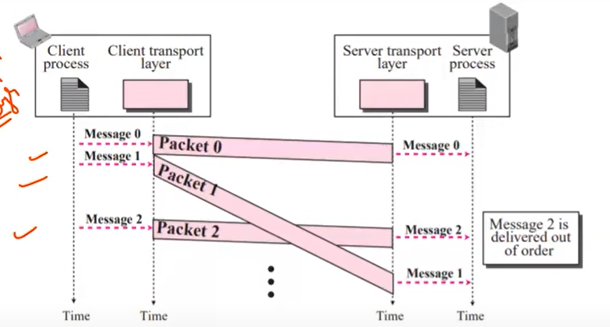
Transport layer adds buffer to sender and receiver side. Buffer is set of memory loc that holds your packets and when it is full it informs your application or sender.

Error control:- Every packet has a **sequence number** so if a packet has been sent again then the receiver knows that it’s a duplicate packet it will discard it. By this receiver will know, which packet is duplicate, which packet arrives out of order, which packet is missing. [other info in header with socket header]

Sequence is 2^m -1

**Acknowledgement:-** Whena packet is send to receiver return an acknowledgement that it has received the packet.

**Connection less and connection oriented services**



In context of transport layer **connection less** means independency of the packets. **Connection oriented** means dependency between packets.

See diagram when connection less service is used packet from transport reaching to the destination in unordered fashion. In this if the packet is lost then we won’t be able to find out which packet is lost coz there is not order, there is no numbering of packet. E.g. UDP

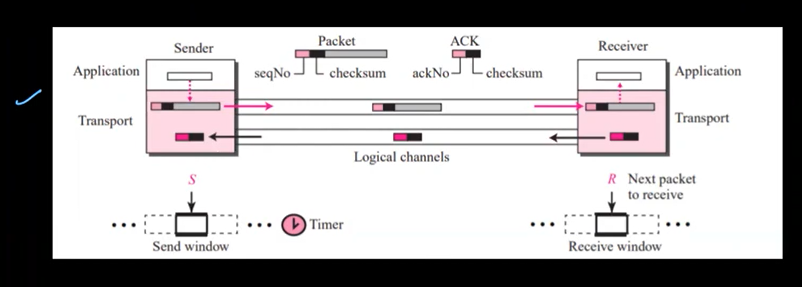
Connection oriented Service :- First connection is made then data is transferred after thant connection is teared down. Data will arrive in order. E.g. TCP

Services done-----------------------

**Protocols**



2) Stop & wait protocol:-

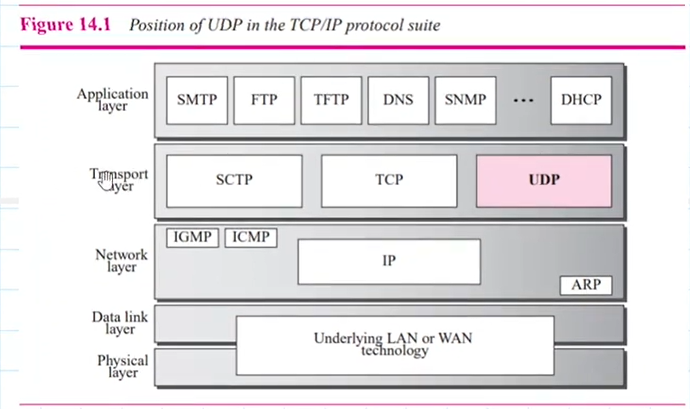


When the packet is received it will send back an acknowledgement. There is a timer for acknowledgement if sender does not receive it then it send the data packet again. Less chance of missing a packet. [There are other protocols as well but are covering less time].

**UDP[User Datagram Protocol]**

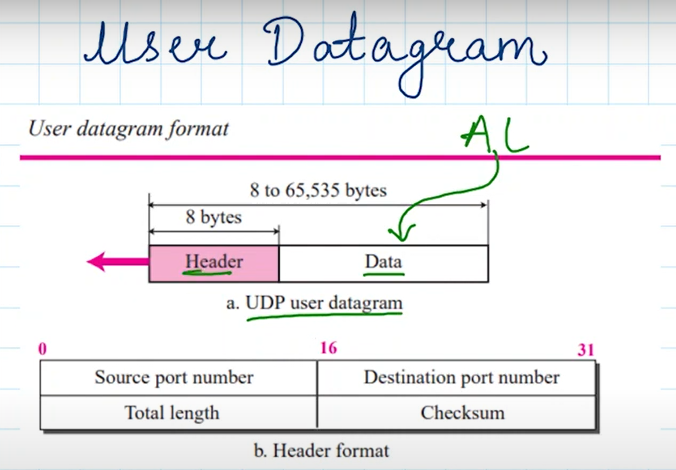
It’s connectionless protocol and unreliable service

There is a trade-off between speed and reliability, you can only one.



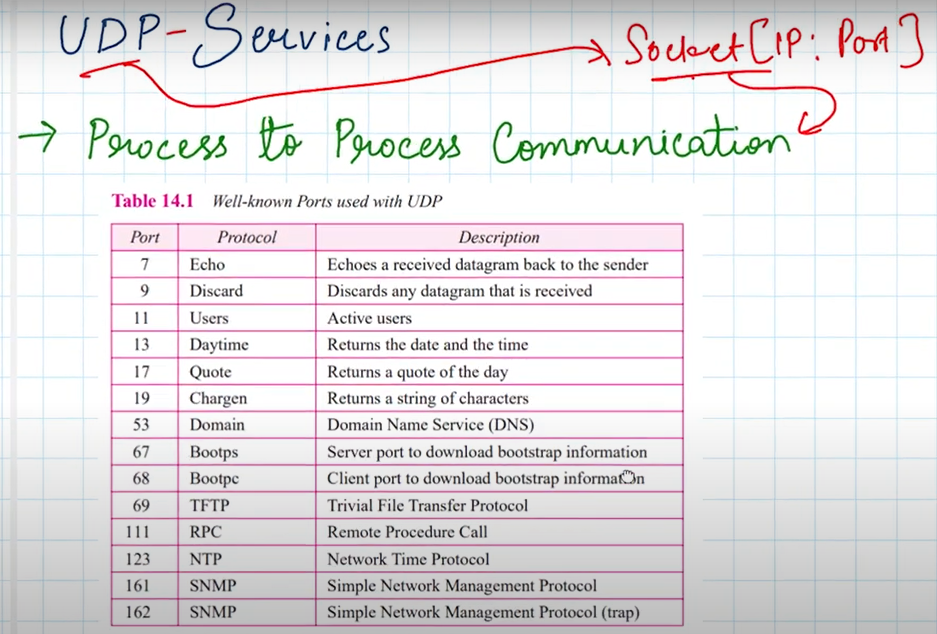
Things to be covered

1. UDP Datagram
2. UDP Services
3. UDP Applications
4. UDP Packages

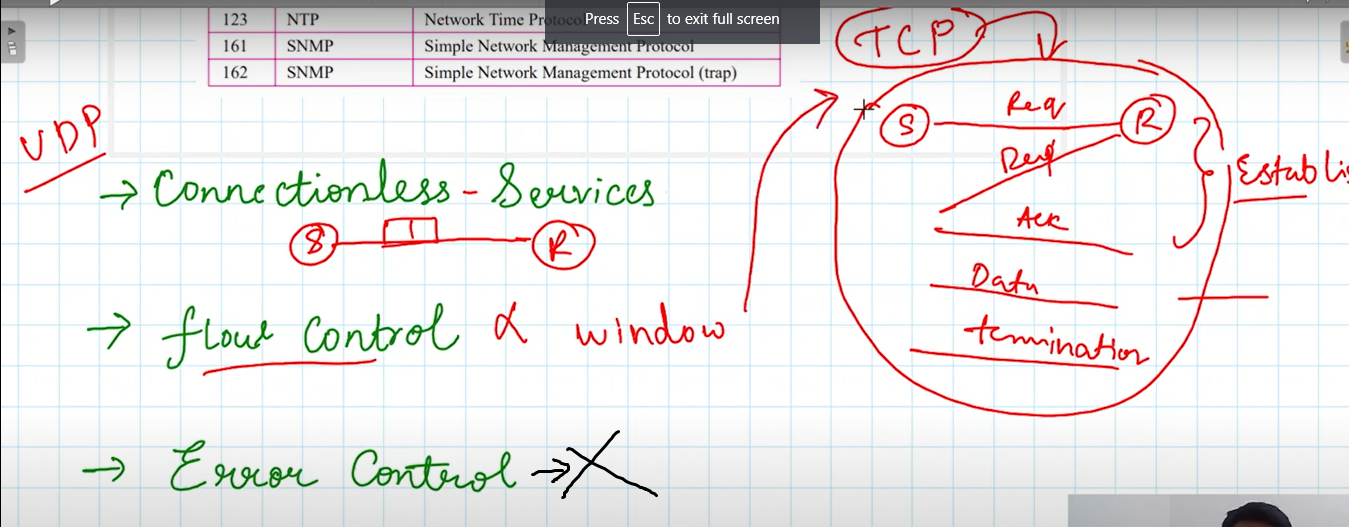


Port num – 16 bit

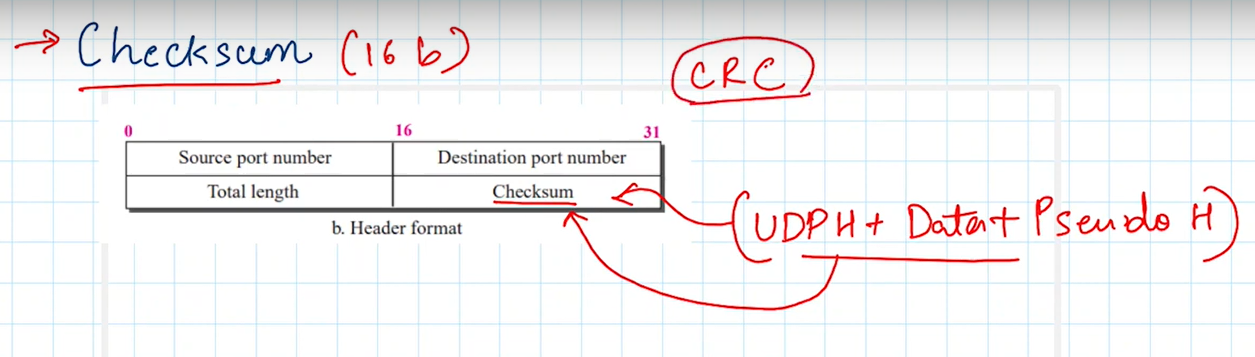
Total length - Total length of UDP Packet, UDP length = I/P length – I/P header

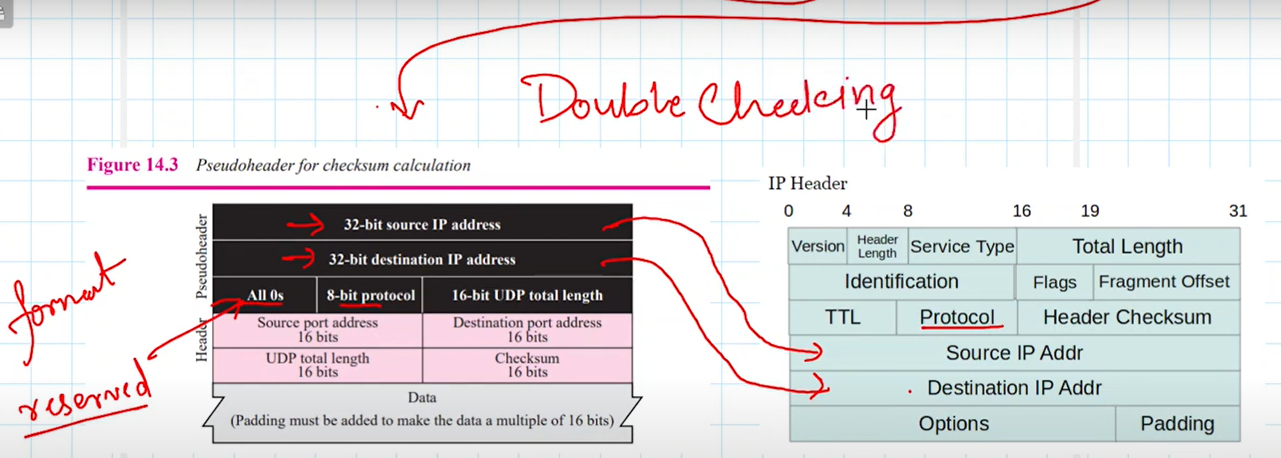


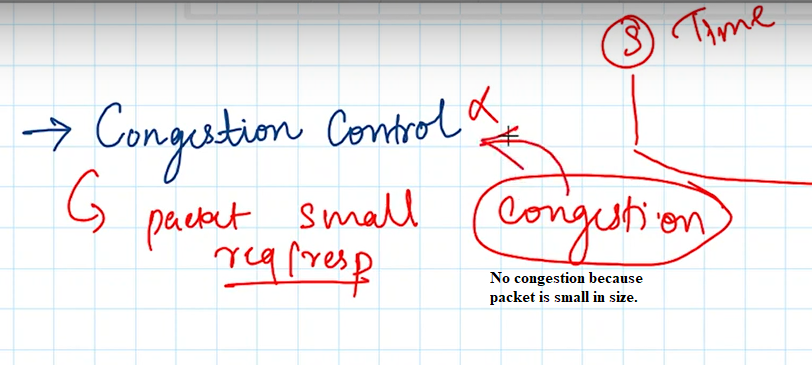
DNS uses UDP, coz this is a short req short response application.

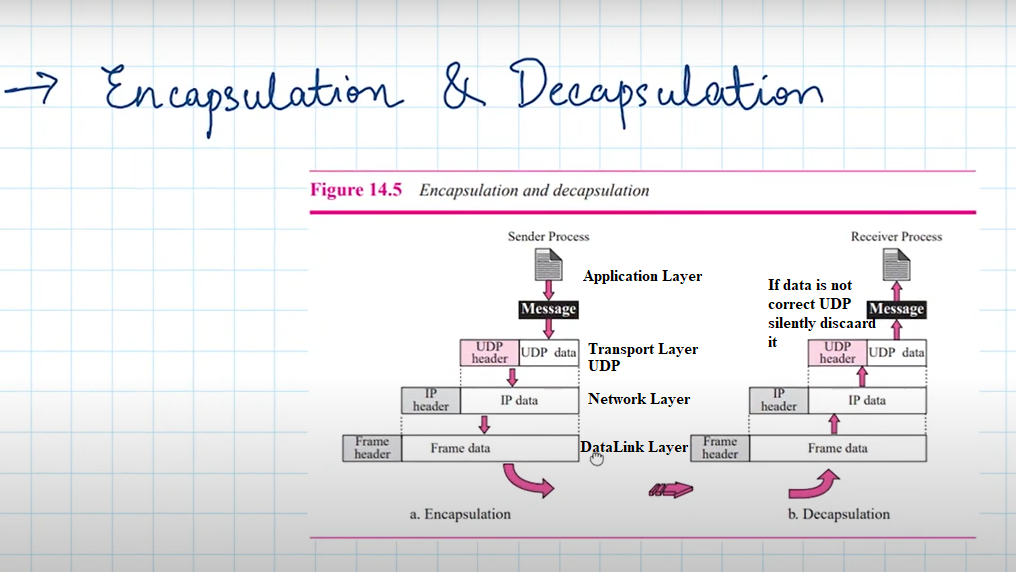


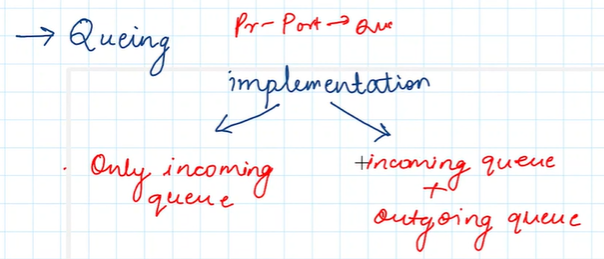
No error control but we have checksum[optional]. Coz checksum is time consuming hence is considered optional in UDP.

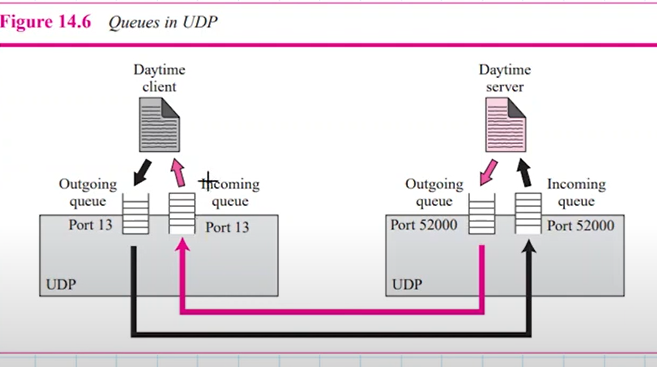




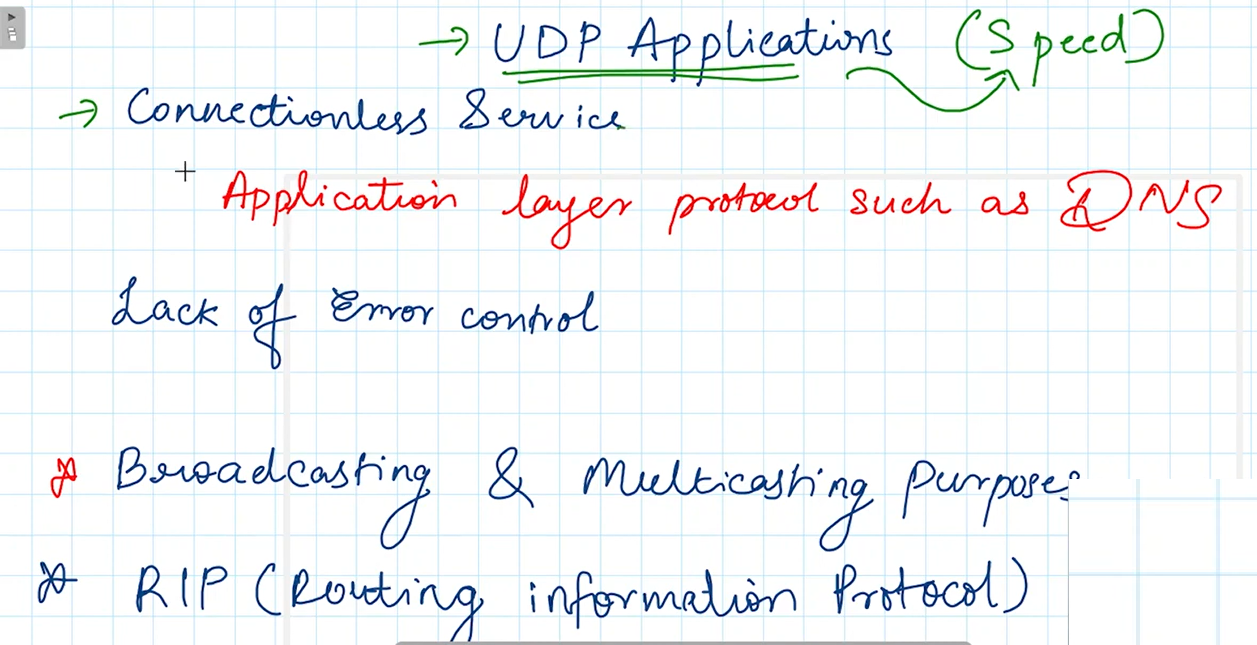








If queue overflows then OS slows down the process



Lack of error control is helpful feature in real-time streaming of videos, audios. Suppose we use TCP instead of UDP for this purpose then if a pixel[Data] is missed then TCP will resend it and till that time you have to wait means whole screen become blank for seconds until the lost frame is not transmitted by TCP. Since we don’t want blank screen we use UDP it keeps sending the data/frames and speed is really high that user might not even recognize a missing frame.

Last Topic is UDP package – Internals of how UDP sends the packets.

<https://www.youtube.com/watch?v=nFQXB3dN7Us&list=PLLiEAVKgMJEP8QrI45r3JatbaP3FuHa2r&index=92>

See this for more info