25/9/19

Date / /

Page No

→ RIP → Routing Information Protocol

In routing table in N/W address first column will have destination ~~add~~ address.

when the ~~th~~ do not know the link the default value is 16. which means . max . no. of nodes connected

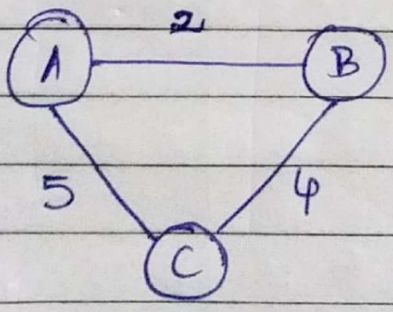
→ Link state algorithm
it is for intra - domain
based on Dijkastra's Algorithm.

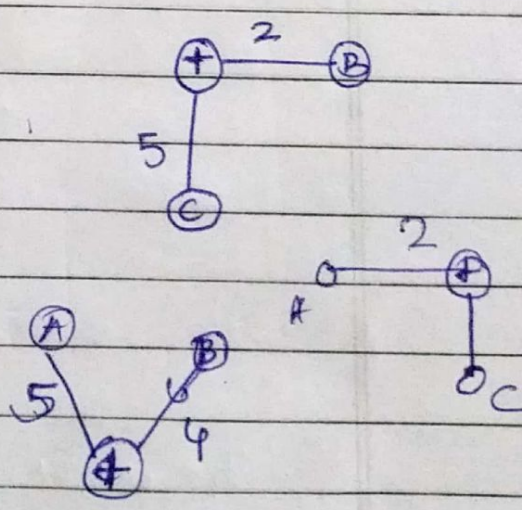
when the nodes connected in the network have all information ~~tha~~ is available of each every node

Open Shortest path first .

the min path is considered ~~from~~ & it is entered

why?
assume all nodes have initially partial knowledge.



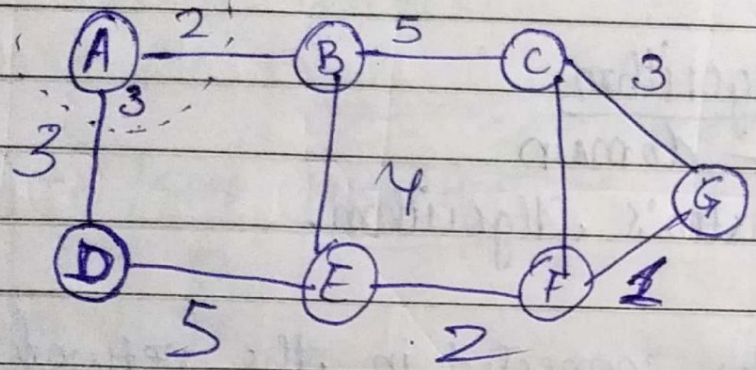


dynamic routing table

(1) for each & every node, link state packet shares.

(2) the packet is shared ~~in~~ with immediate next node

the ~~flooding~~ flooding

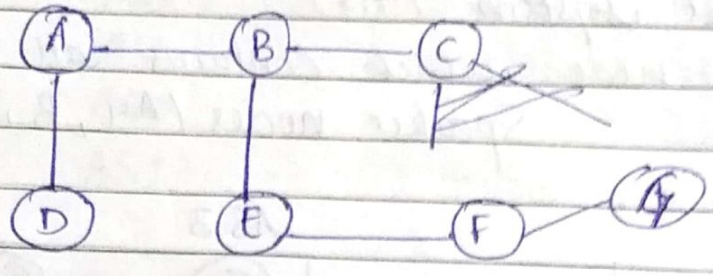


link State packet

the immediate nodes of the a particular node the cost is calculate

after

Shortest path



Routing table
node A

To	Cost	Next
A	0	—
B	2	—
C	7	B
D	3	—
E	6	B
F	8	B
G	9	B

27/9/17

DINKY

Date: / /

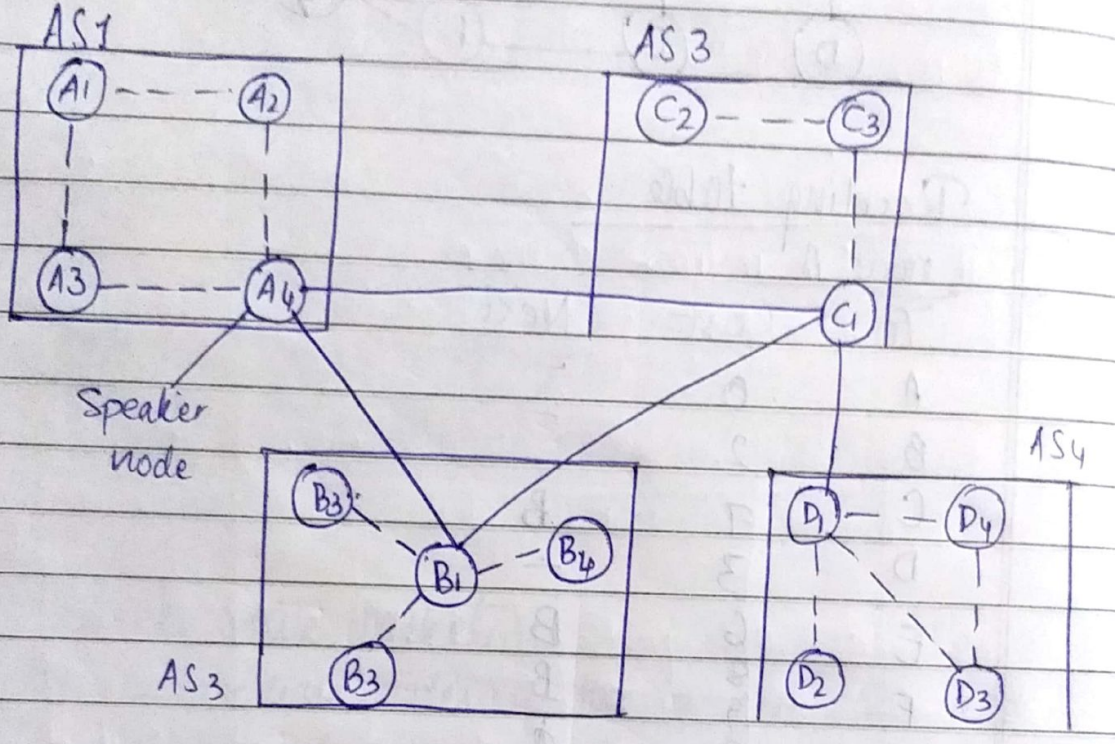
Page No.

Path vector Routing (Inter-Domain Protocol)

Speaker node

autonomous system (AS)

Assuming a node which contains all info. of Speaker nodes (A_4, B_1, C_1, D_1)



Speaker nodes of diff. n/w are connected with each other for passing info.

AS1 n/w has 4 nodes & will be in.

Routing table will be of the speaker nodes.

Speaker nodes have right to pass on info.

A4 Routing Table

Dest.	Path
A1	AS1
A2	AS1
A3	AS1
A4	AS1

C1 Routing table

Dest.	Path
C1	AS3
C2	AS3
C3	AS3

B1 Routing table

Dest.	Path
B1	AS2
B2	AS3
B3	AS3
B4	AS3

D1 Routing table

Dest.	Path
D1	AS4
D2	AS4
D3	AS4
A4	AS4

after initial routing table we have sharing

A4 will share its routing table to C1 & B1

C1 ⟶ A4, B1, D1

if A1 nodes wants to send mode to B1 then we need to update RT the new entries will be added into its RT.

A1

Dest.	Path
A1	AS1
⋮	⋮
A4	AS1
B1	AS1-AS2
⋮	⋮
B4	AS1-AS2
C1	AS1-AS3
⋮	⋮
C3	AS1-AS3
D1	AS1-AS3-AS4
⋮	⋮
D4	AS1-AS3-AS4
A1	

B1

Dest.	Path
A1	AS2 - AS1
:	A
A4	AS2 - AS1
B1	AS2
:	:
B4	AS2.
C1	AS2 - AS3
:	
C3	AS2 - AS3
D1	AS2 - AS3 - AS4
:	
D4	AS2 - AS3 - AS4

C1

dest.	Path.
A1	AS3 - AS1
:	:
A4	AS3 - AS1
B1	AS3 - AS2
:	:
B4	AS3 - AS2
C1	AS3
:	
C3	AS3
D1	AS3 - AS4
:	
D4	AS3 - AS4

D1

dest.	Path.
A1	AS4 - AS3 - AS1
:	:
A4	AS4 - AS3 - AS1
B1	AS4 - AS3 - AS2
:	:
B4	AS4 - AS3 - AS2
C1	AS4 - AS3
:	:
C3	AS4 - AS3
D1	AS4
:	:
D4	AS4.

Page No.

Order Gateway protocol

Ch-5

28/9/19

Transport Layer

→ Process-to-process comm. → process-to-process communication.

→ UDP (User Datagram Protocol) (from source system.

diff b/w UDP & TCP

30/9/19

→ TCP — Transmission Control Protocol)

connection-oriented & reliable

flow & error control is managed.

Diff. UDP & TCP — connection-oriented.

~~UDP — Contionless~~ Connectionless

Unreliable

UDP & TCP. → reliable

Connectionless Connection Oriented

In UDP the upper layer data converted in datagram

TCP in segments

UDP	TCP
no flow control	flow and error
header → 8 bytes	heade → 20-60 Bytes
upper-layer → datagrams	Internet upper-layer data → segments
unreliable connectionless	reliable. Connection-oriented

In process-process communication different ports are used.

(i) Well know ports → ~~0-2, 10-2~~

If port numbers are dynamic.

stream

→ ~~String~~ delivery service

allows sending process to deliver data in stream of bytes & ~~data is sent in string form~~

allow receiving process to obtain stream of bytes

~~In virtual~~

TCP creates a virtual Connection in which 2 process ~~are~~ are seem to be connected by imaginary tube carring data.

In a system, there are multiple processes going on. so,

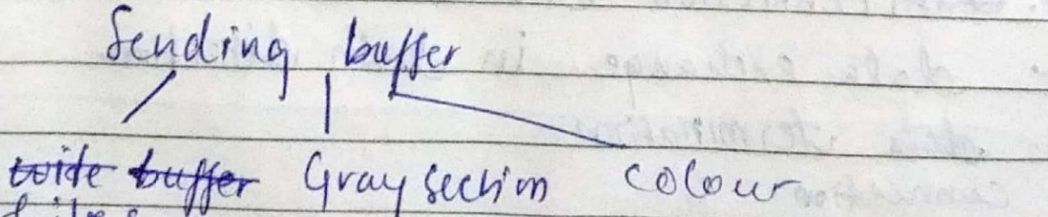
→ Sending & Receiving Buffer.

TCP needs buffer for storage.

receiver have buffer which ~~used~~

Sending buffer → receiving buffer

Senders & receiver have buffer which will store the date which is used in flow control.



Gray → Holds the byte that have been send but not acknowledged. TCP keeps until ack is received

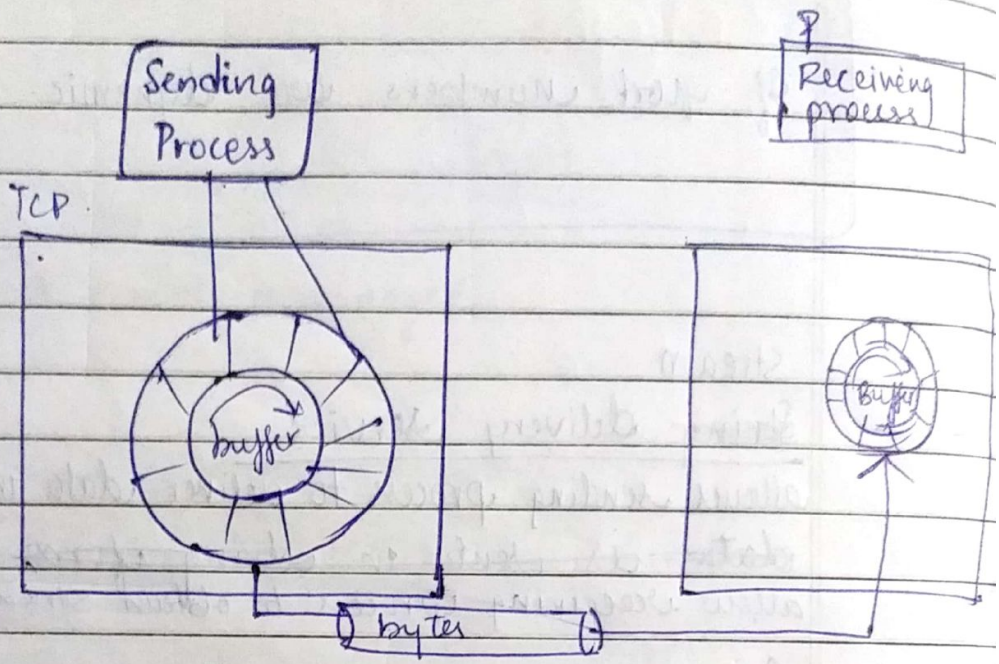
if process

~~Extra bytes~~ which blank space - White.

If bytes send and no ack - Gray.

One by one byte is sent → colour ←

It contain byte to be sent by sending TCP



at receiving ~~section~~ side

(i) White section ← empty chambers; filled by bytes receive from network.

(ii) colour section ← If receiver not received. contains received bytes; once read chamber is recycled; empty

Segments → The data ~~is~~ in colour section is converted in segments.

Segment size differs.

→ Full duplex Communication

In TCP - full duplex communication

1. ~~Conti~~ connection establishment
2. data exchange in both direction.
3. ~~data~~ termination - connection

4. Reliable server

→ Numbering system

(i) sequence no.] - byte no., sequence no, ack. no.
(ii) ack no.

Segment is collection of bytes

byte no. → no. of bytes in the segment which have a random number. is known as byte no.

sequence no. → Segment no. assigned.
after bytes have been numbered, TCP assigns seq. no.

The first byte no. is considered as sequence no.

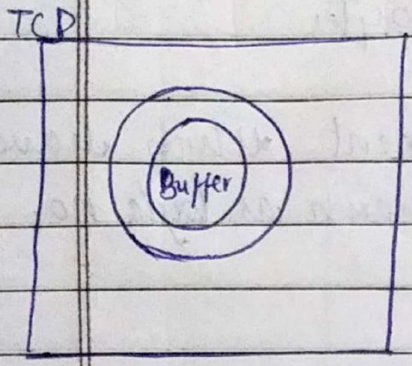
ack no → ack no is sequence no. +1.
(next byte no.)

Q: TCP connection is transferring a file of 5000 Bytes. The first byte is number 10001 what are the sequence numbers for each segment. if data are sent in 5 segments, each carry 1000 Bytes

10001 Segment: 10001.

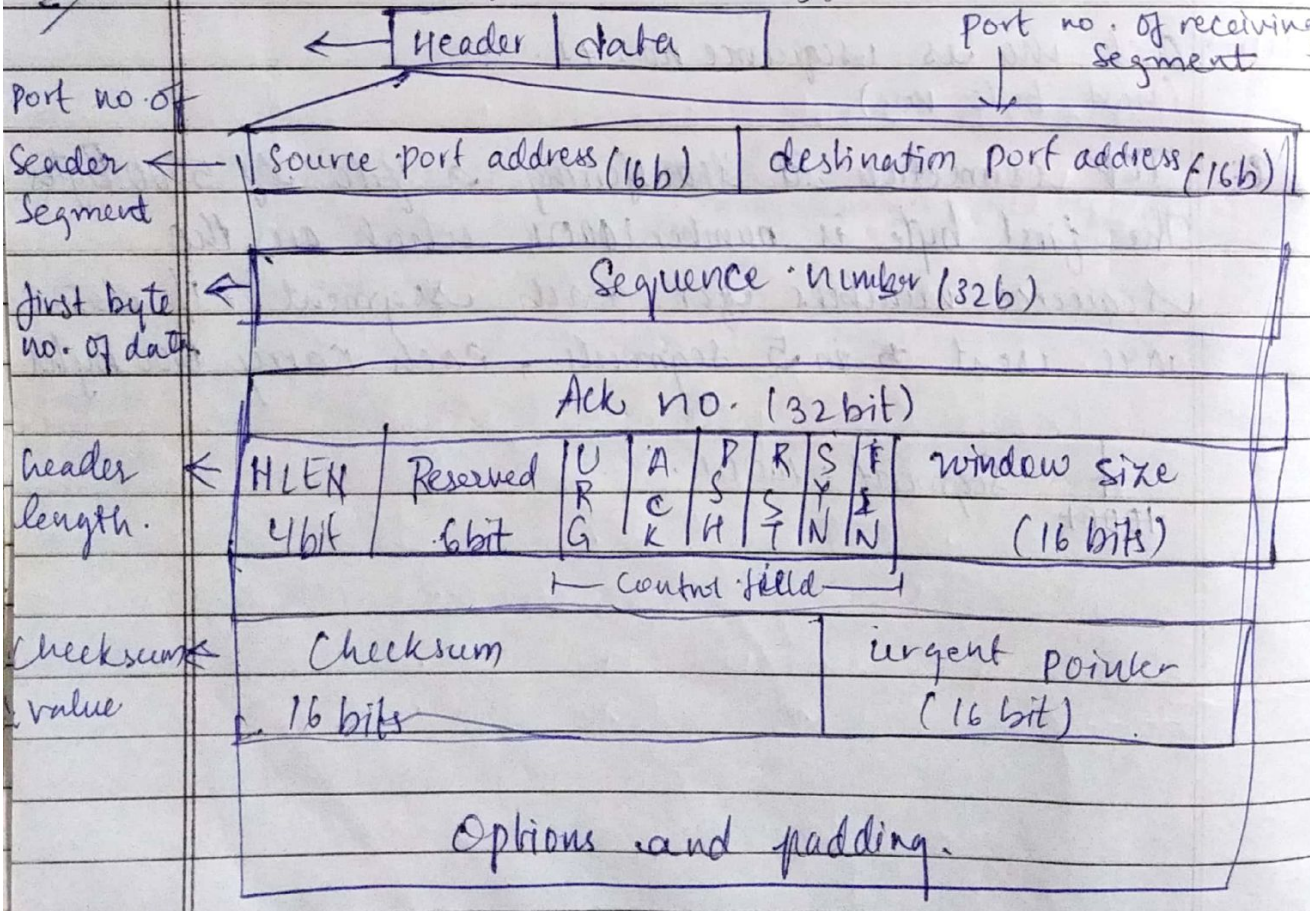
1/10/19

Sending process



Segment : Format ⟹ packet in TCP called segment

→ 20 – 60 bytes



Header length → No. of 4 bit word in TCP header.

DONKY
Date: / /
Page No.

urgent pointer → if set that particular

Sender port address → is needed for process to process communication

Sequence number → for flow control.

ack no. → byte no. of the next

6 flags → urgent flag
Synchronise flag
push flag
finish flag
reset flag
terminate

Options & padding → segment size

URG → Urget pointer feild is valid
ACK → ack value is valid
PSH → push the data
RST → Reset ~~Corn~~ connection
SYN → Synchronize sequence no. during connection
FIN → terminate the connection

Control field

URG	ACK	PSH	RST	SYN	FN

- URG → for priority
- PSH → for data transfer PSH=1
- RST → RST Connection
- SYN → Synchronise sequence numbers.
- FN → Connection terminated

→ TCP Connection

3 phases.

1. Connection establishment :
   → 3-way handshaking
2. data transfer
   → after connection established
3. connection termination
   → 3-way handshaking

Date: / /
Page No.

→ Connection establishment using three-way handshaking.

The time when requesting for client for communication then it

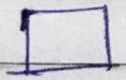
A. Ack flag
S. SYN flag

firstly
Connection establish. ~~then~~ for that it will request client server by setting SYN flag then server will check if free then server will send ~~step~~ random sequence no. and ack is also sent

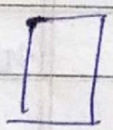
client will send ack. acks for the same sequence no.

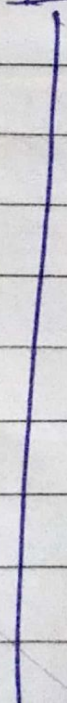
Client → Server
Server → client
Client → Server
} three way.

client



server



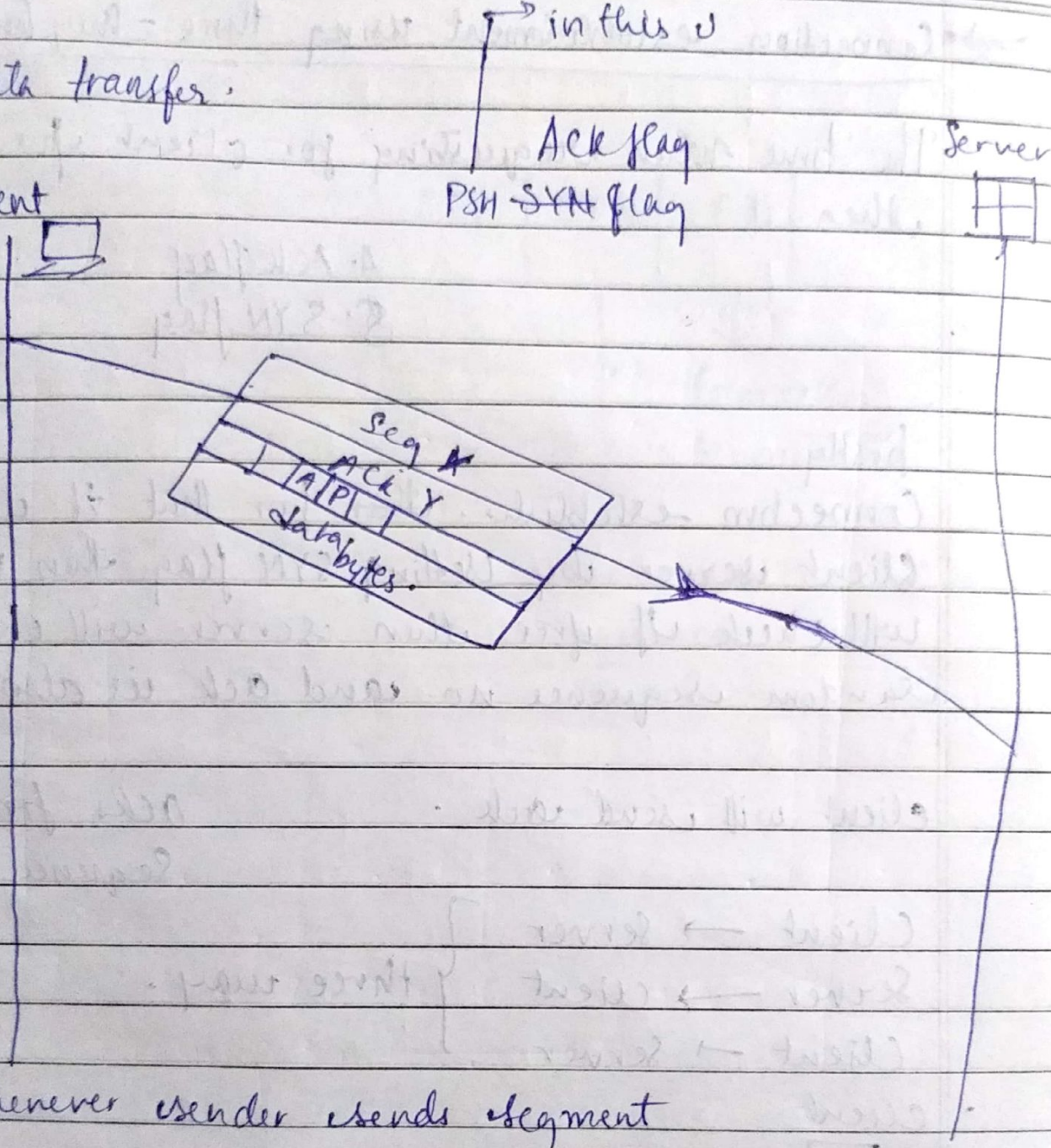


PSH. if set ~~send~~ receiver will consider that that particular segment has has data bytes to be received.

DINKY
Date: / /
Page No.

in this ↓

data transfer.

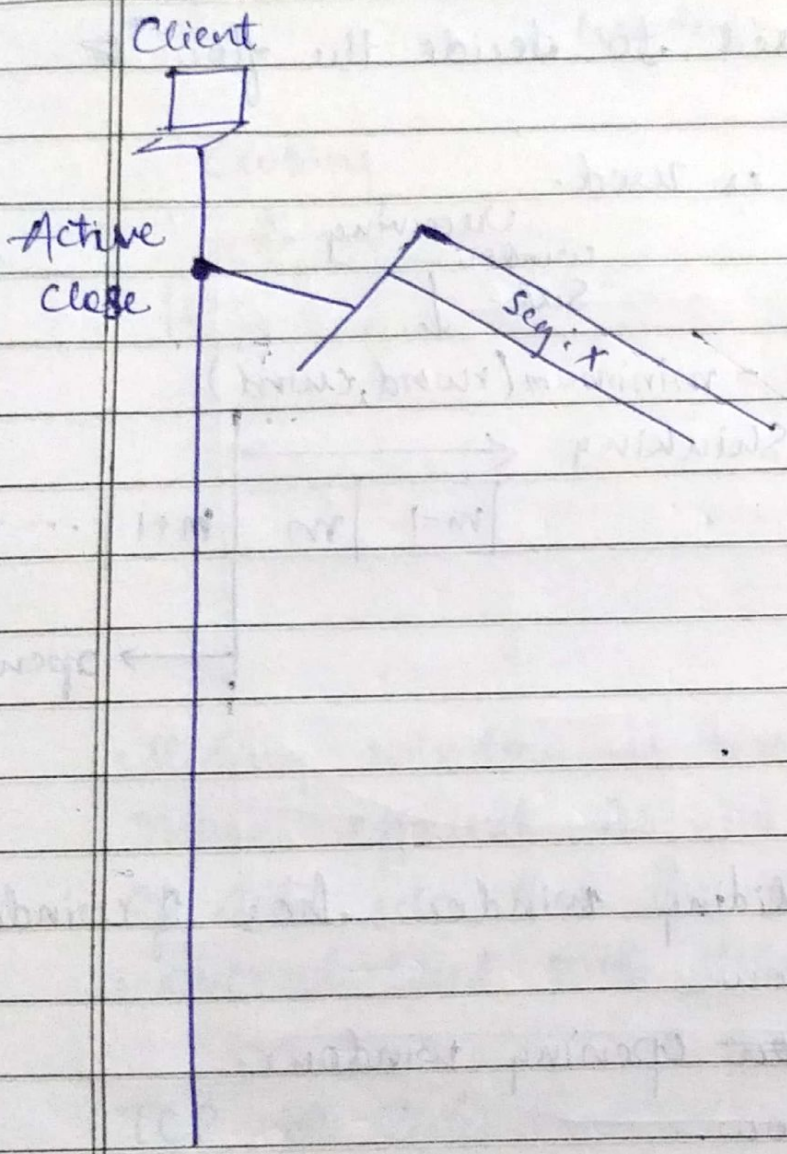


whenever sender sends segment

3/10/19

Explain Connection establishment or termination.
Explain the procedure three-way handshaking.

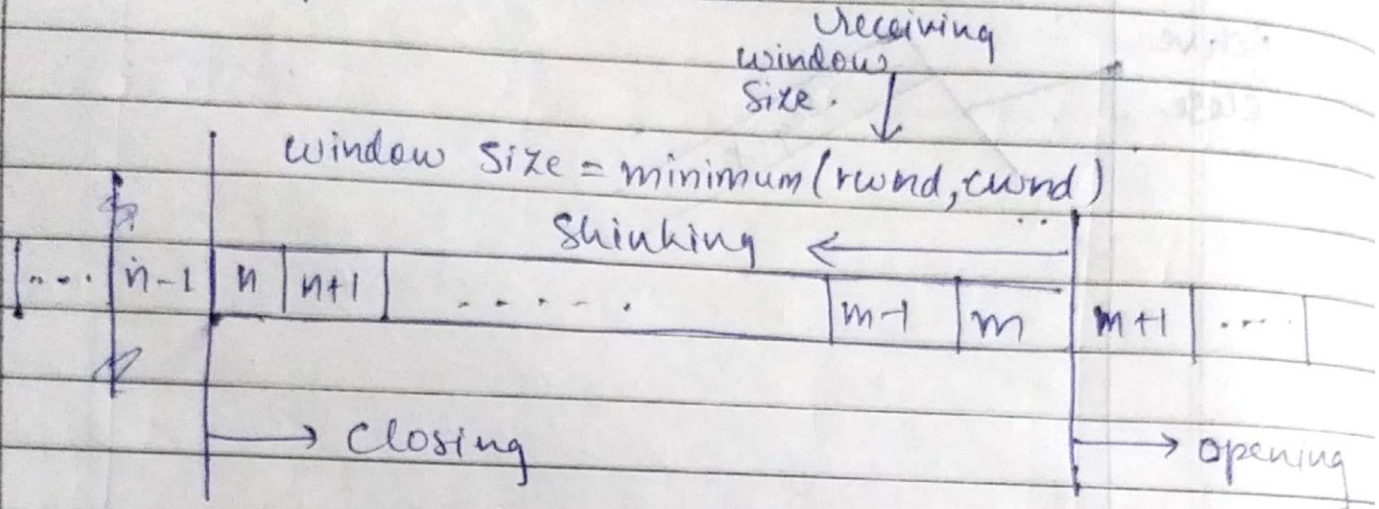
→ Connection Termination using three way handshaking



Date: / /
Page No.

Flow Control

flow control is used to decide the flow of data. Sliding window is used.



In flow control sliding window has 3 windows.

1. Shrinking window
2. ~~Sliding window~~ Opening window.
3. Closing window.

all windows are managed by receiver & Sender have to do to obey the instructions received from the receiver

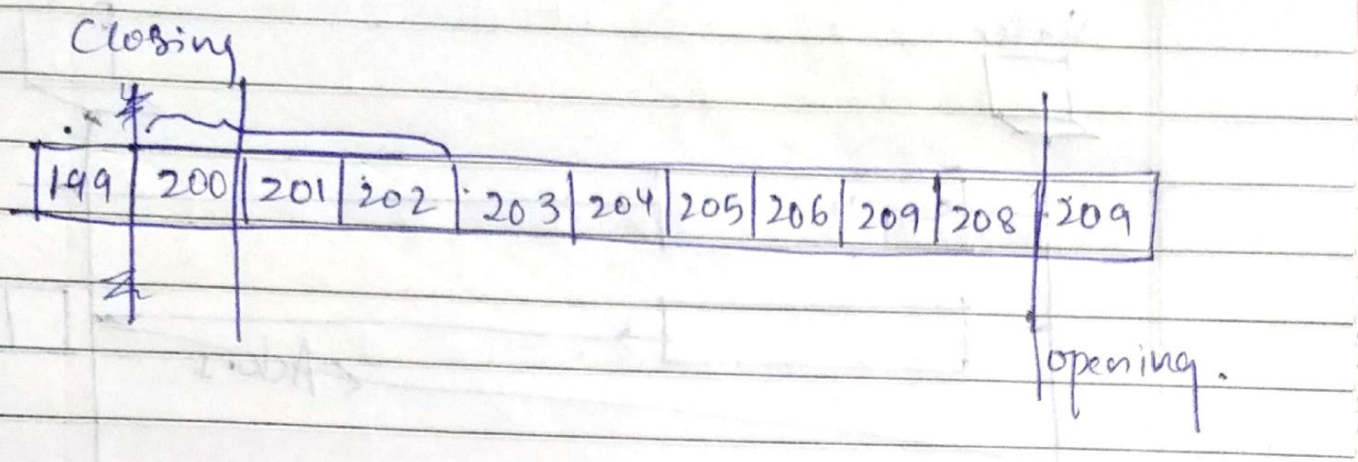
Pg 731
~~739~~

Opening window allows the new bytes into buffer.

fig 23.23

They remain in buffer till ack not received.

Window size = Min(20, 9) = 9.



Sliding window is used to make transmission more efficient as well as to control the flow of data so that the destination does not become overwhelmed with data

TCP ~~1st~~ sliding windows are byte-oriented.

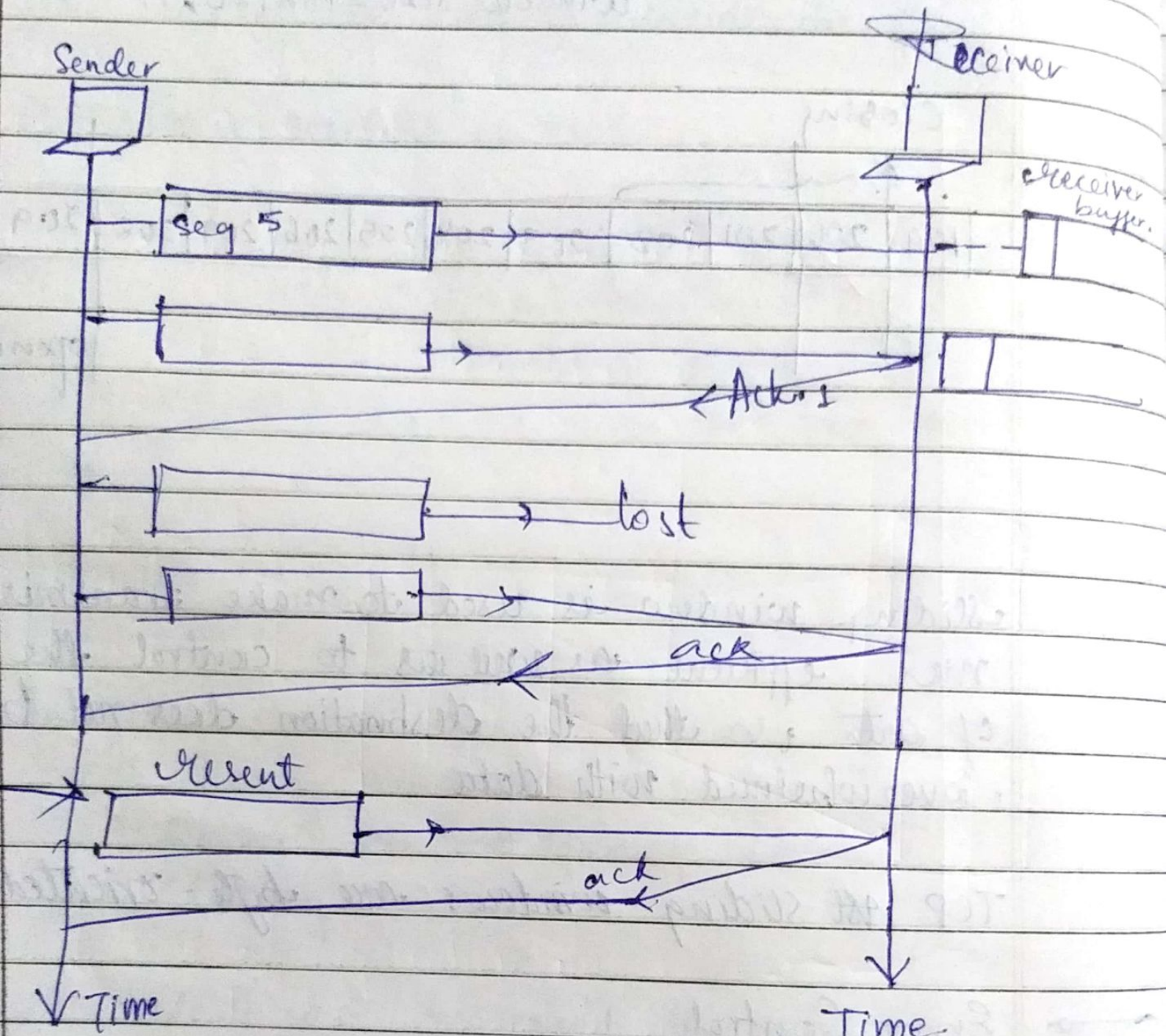
→ Error Control

Retransmission:

Retransmission after Run time Out:

1.
2.
3.

Lost Segment :



Unit-6

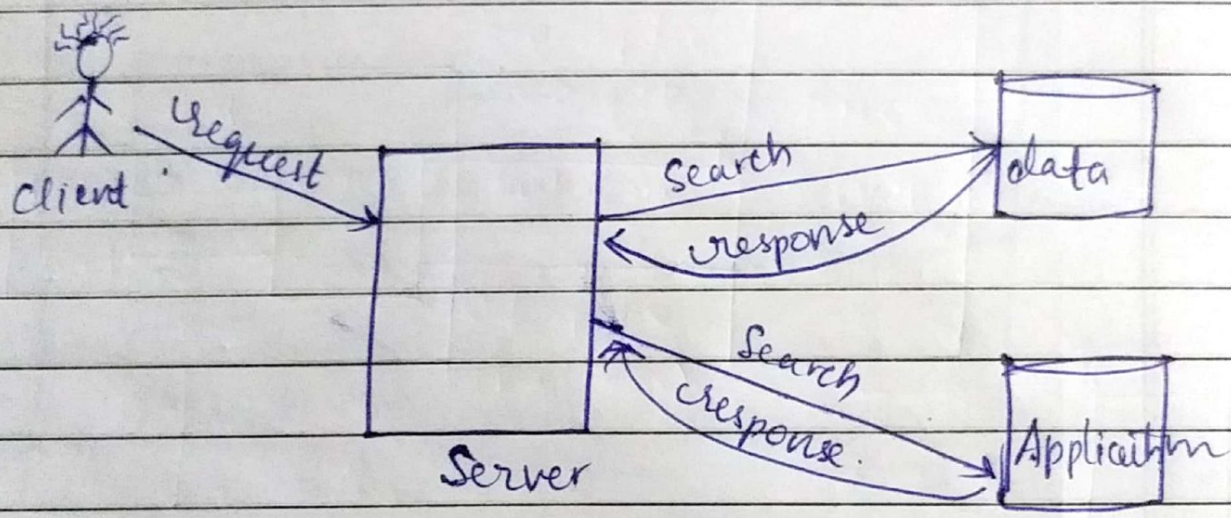
Application layer

Top-most layer of TCP & OSI model

used for providing services, ~~sets~~ security
Network Connectivity

intermediate layer b/w User & transport layer

always follows - client - server architecture



Application of application layer:

1. E-mail.
2. Youtube
3. Social Media
4. web pages
5. News Channels.

Page No.

Encryption : to provide security. plain text converted into cypher text

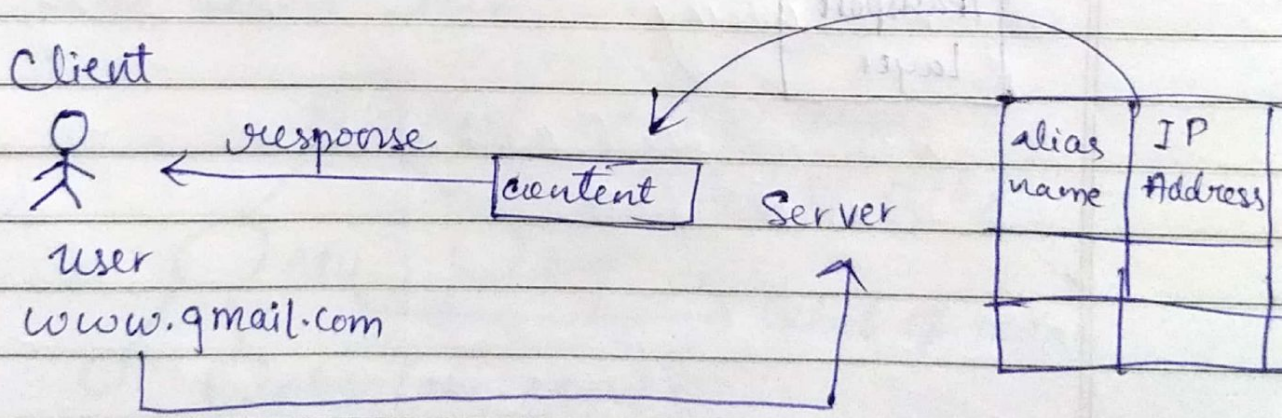
decryption :

15 | 10 | 19

→ DNS → Domain Name System

DNS is used to map an alias address to the IP address. Application programs generally used DNS ~~for~~ mapping.

Host name considered as alias.

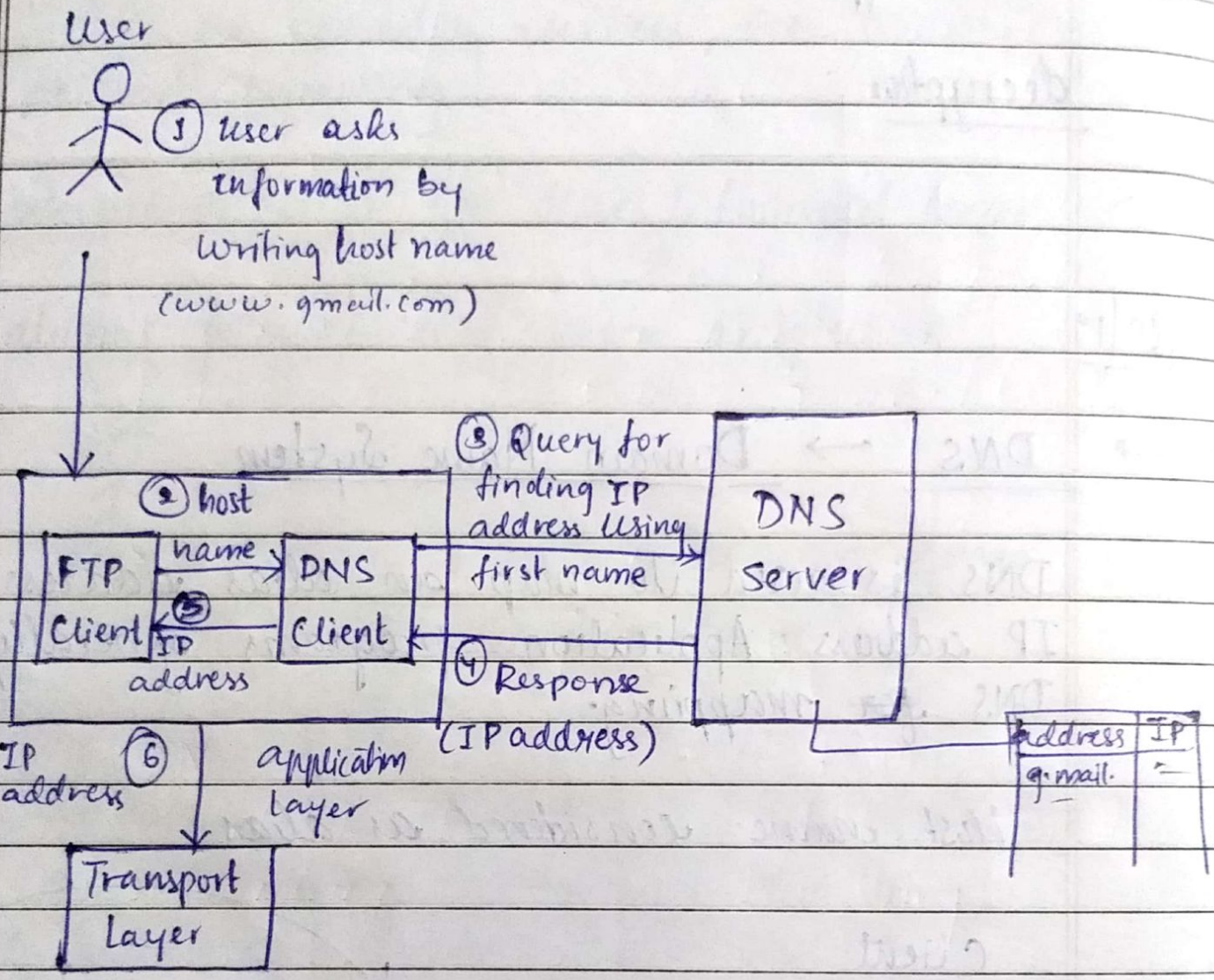


all info is stored in different ways.

It will divide info in multiple parts & stored in multiple systems & these are called DNS systems

The nearest server gets.

Working of DNS :



Namespaces are of two types.

1. ~~flat~~ Flat namespace
2. Hierarchical namespace – for creating domain a structure must be followed.

namespace matches to a unique namespace

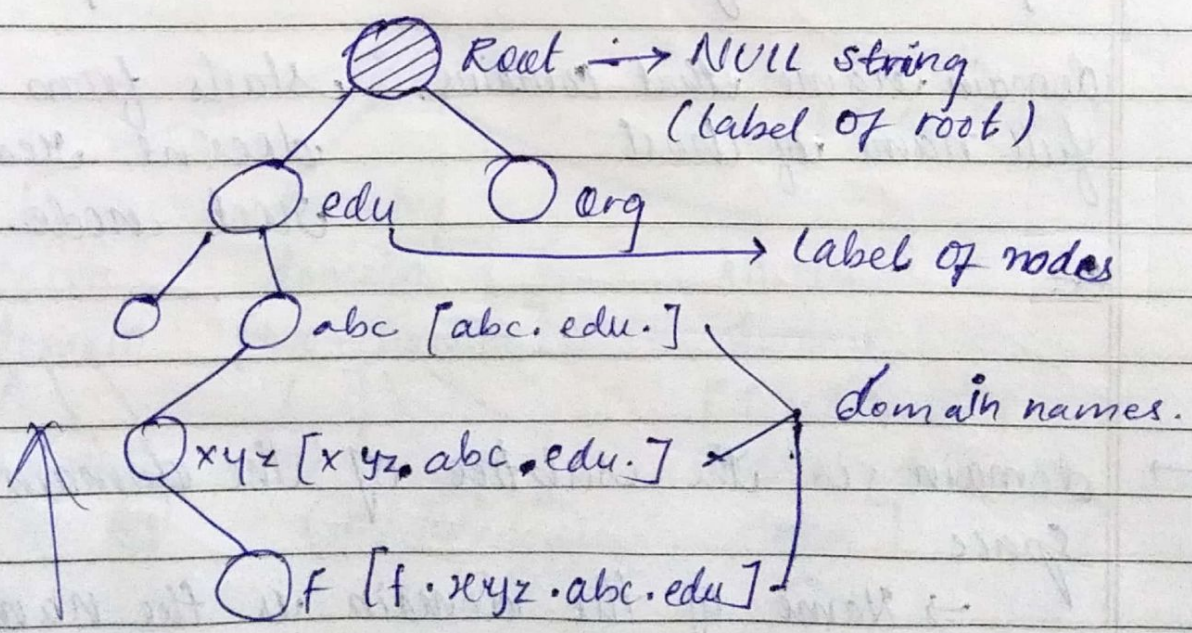
~~DNS~~

→ Domain Name Space

It is designed to have a heirarchical namespace. In this names are defined in inverted tree structure with the root at the top.

A tree can have only 128 ~~neighbour~~ levels.

63 max. level size



A full domain name is sequence of labels separated by dots.

domain names are always read from the ~~root~~ node till root

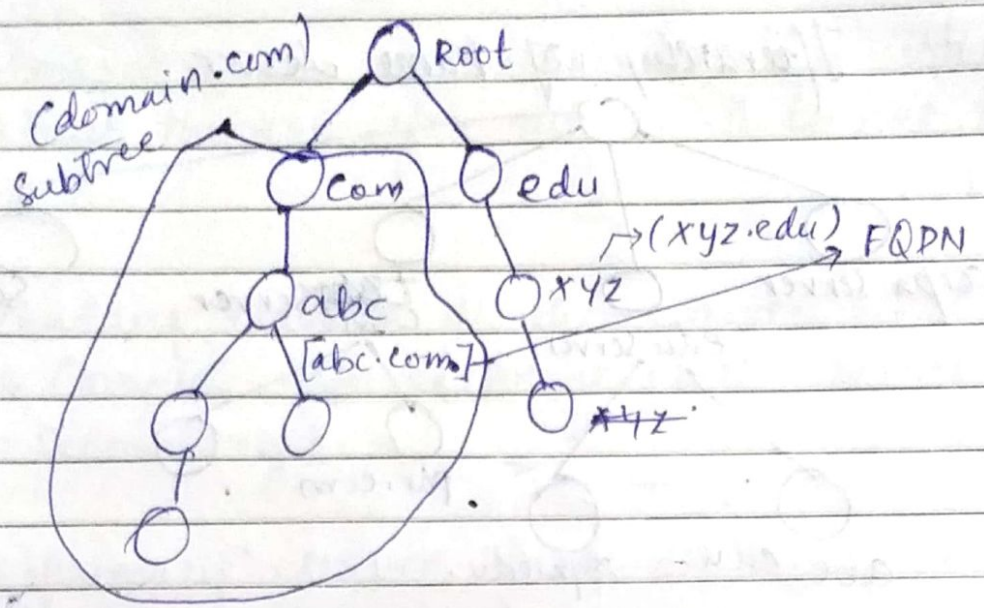
domain name is divided into two parts
(i) fully qualified domain name
(ii) partially qualified domain name. [PQDN]

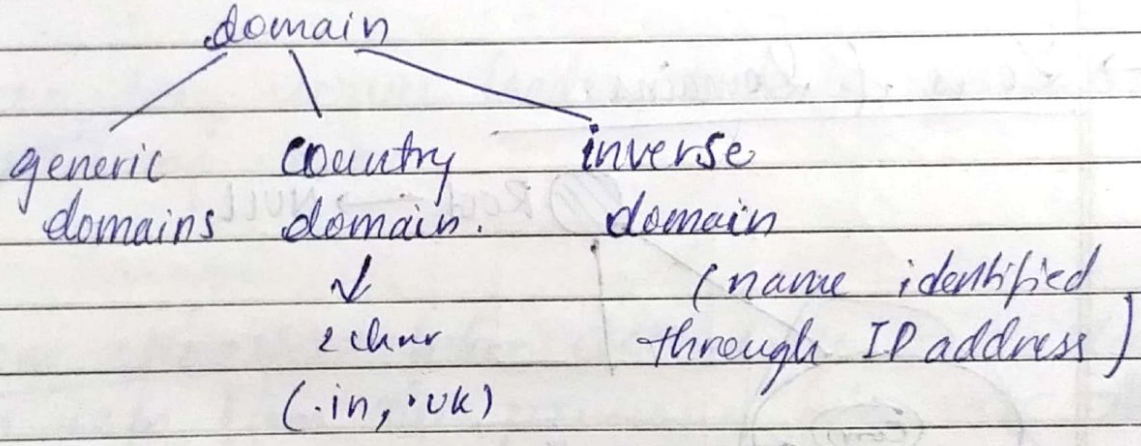
→ always ends with null string
dot is null string (root element)

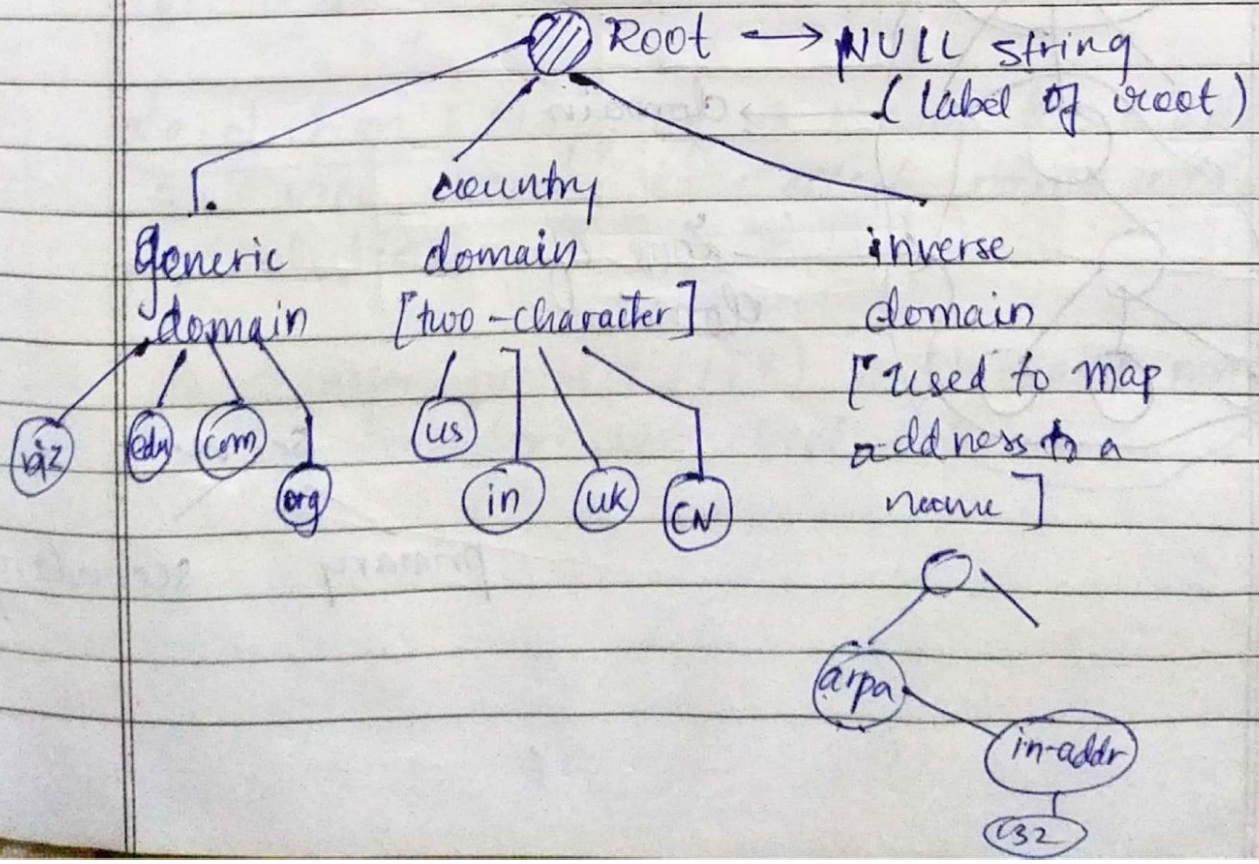
FQDN	PQDN
In this label terminated by null string	not terminated by null string
domain name that contains full name of host	starts from node but doesn't reach till root node.

~~Ex:~~

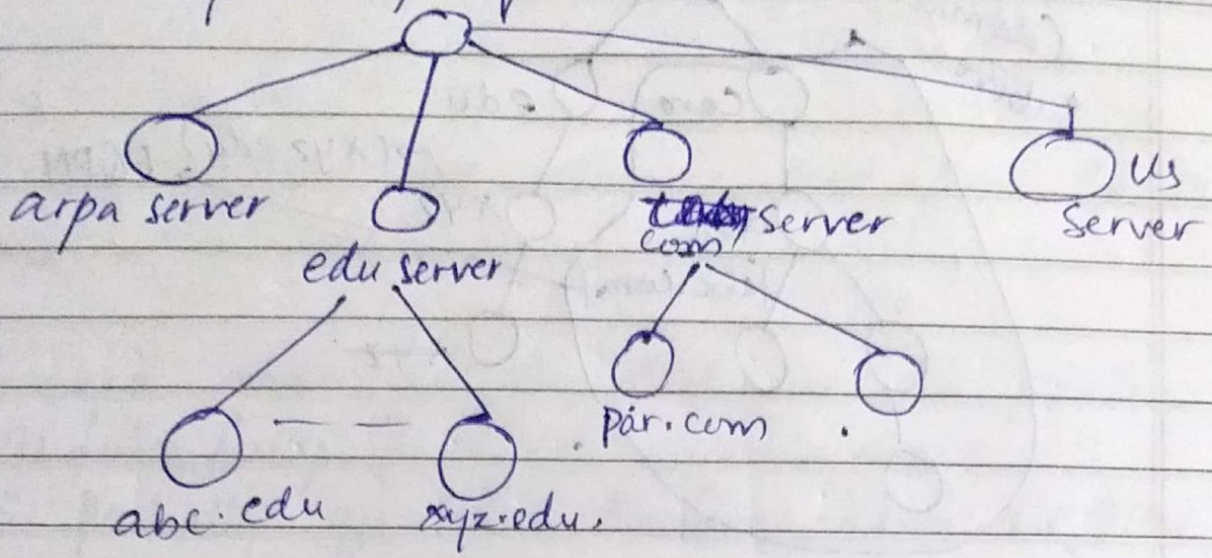
→ domain is the subtree of the domain name space
→ Name of the domain is the name of the node at the top of the subtree



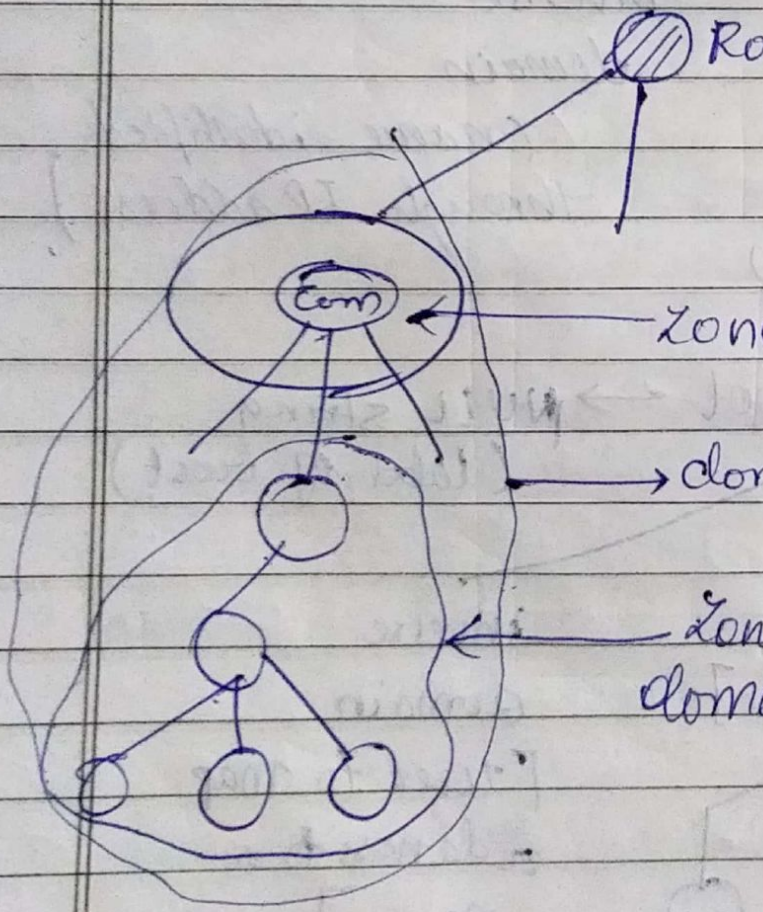




Hierarchy of name server



→ Zones & Domains



Server

primary secondary

→ primary server is the server that stores a file about the zone for which it has an

→ Secondary server is the server that transfers the Complete info from another server (primary or secondary).

a primary server loads all info from the disk file.

Secondary server loads all info from the primary server

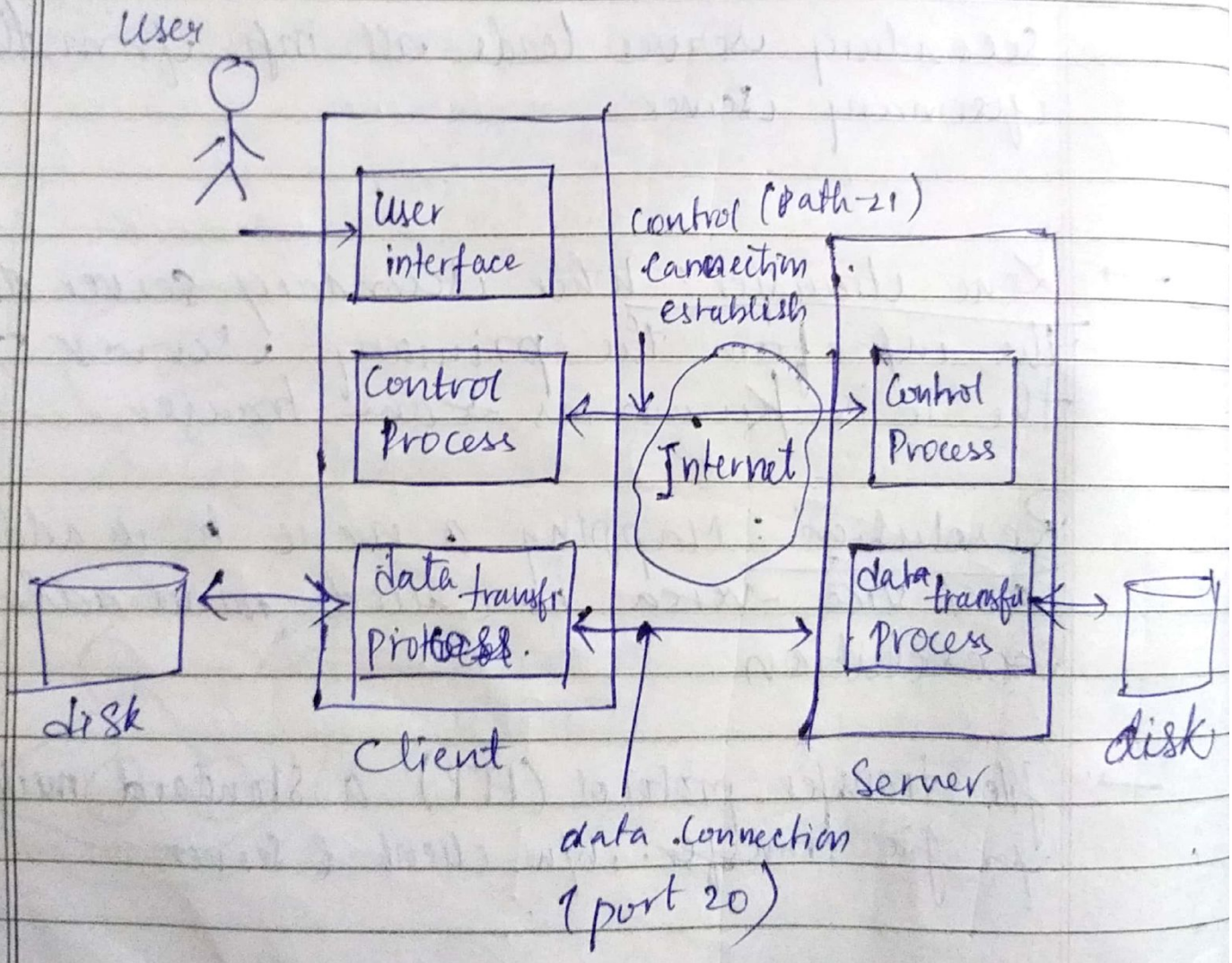
→ Zone transfer: When secondary server downloads the info from the primary server the it is known as zone transfer

Resolution: Mapping a name to a address or vice-versa is called name addres resolution

→ file transfer protocol (FTP) a standard mechanism for file transfer. b/w client & server.

FTP uses 20 port for data transfer
21 for control info: passing connection
Or,

FTP has 2 types of connections.
(i) Open & close for each file transfer.
(ii) Control info connection.
(iii) It remains connected through the entire FTP session

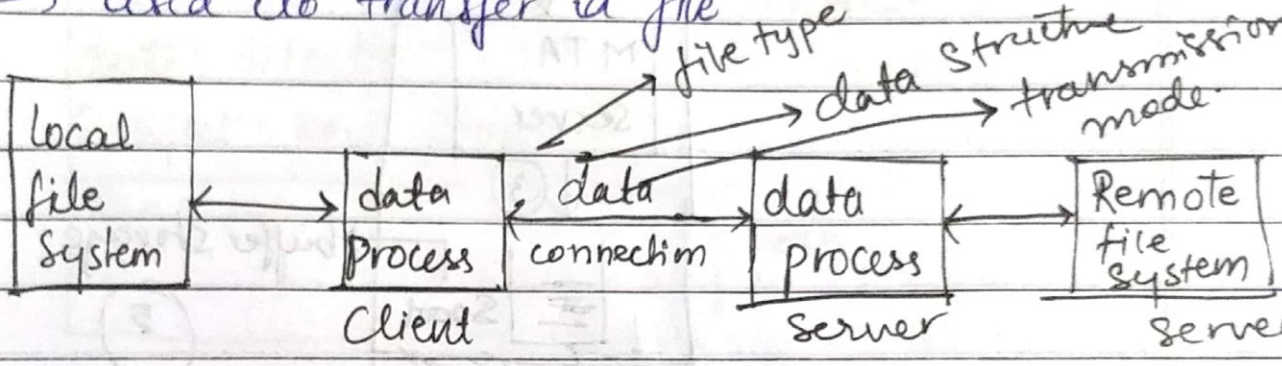


Date: / /
Page No.

Communication in FTP:

Communication over control connection
→ FTP uses a set of ~~the~~ ASCII characters to communicate across control connection.

Communication over data connection.
→ used to transfer a file



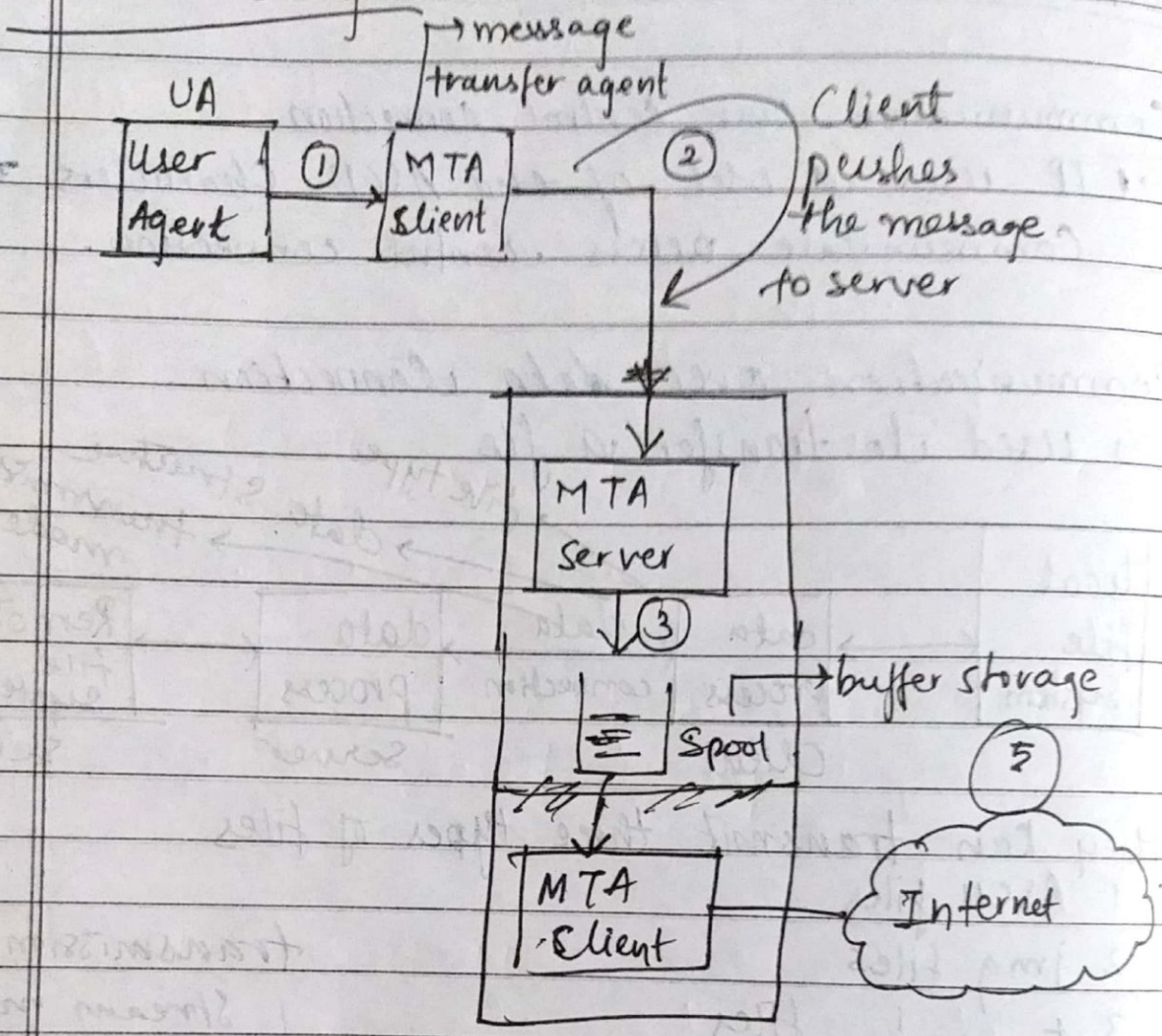
they can transmit three types of files
1 ASCII files
2 img files
3 t l files

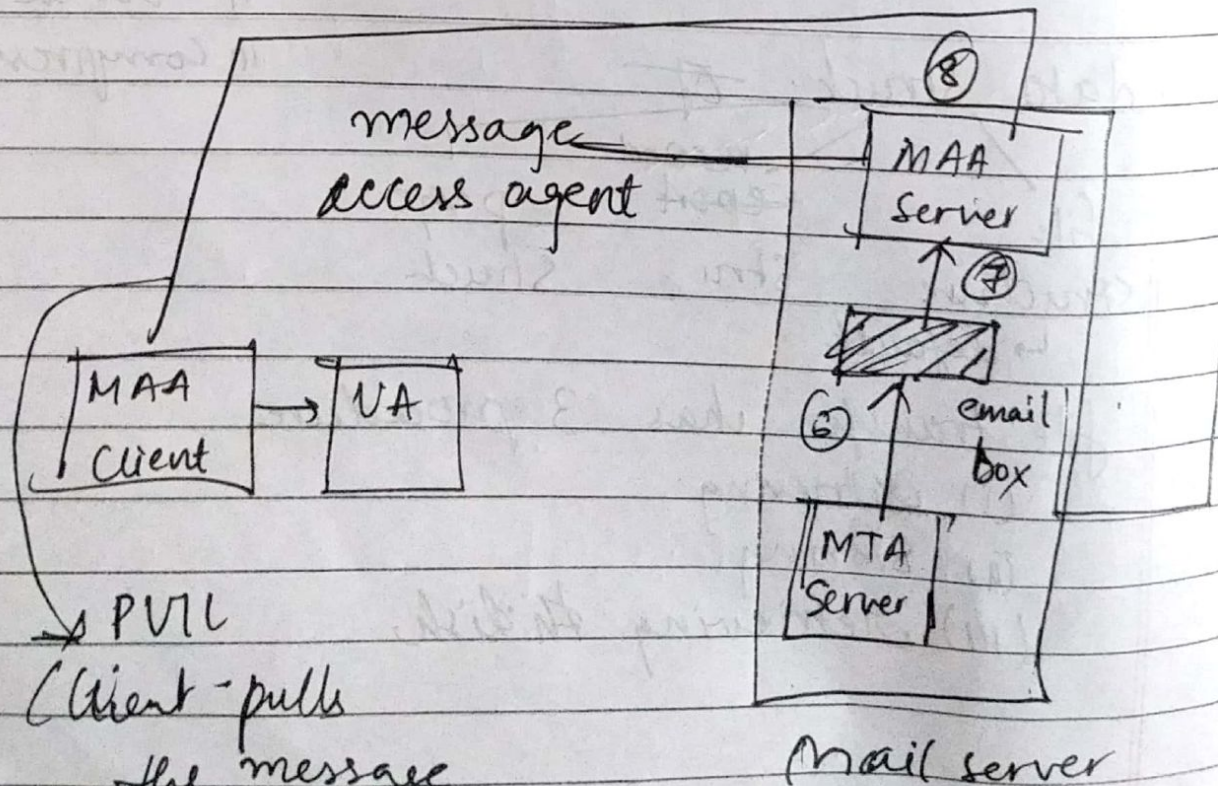
transmission mode
1 Stream mode
11 block mode
111 Compress mode

data struct. ~~of~~
- file structure → default
- record ~~report~~ stru
- page struct

file transfer has 3 procedure
(i) retrieving
(ii) Storing
(iii) retrieving th dirh

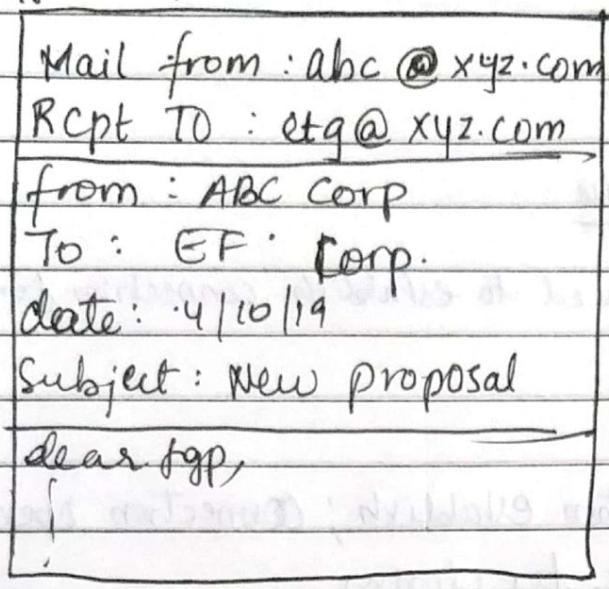
→ Email Diagram





User agents provides user interface

Mail format



Diff. HTTP & FTP

7 | 10 | 19

HTTP

generally used to access WWW or web pages which are already hosted on.

port 80 to access web pages.

get → access data (doc) from server
post → Send info from client to server
head
put → request of some doc from

Command

HTTP Status Code

~~Error~~ 404 → doc. not found

403

2 types of connections

(i) persistant ← default → need to establish connection for each & every file

(ii) non-persistant

request ; connection establish ; connection open till time out or request for closing

→ before HTTP 1.1. ; connection establish