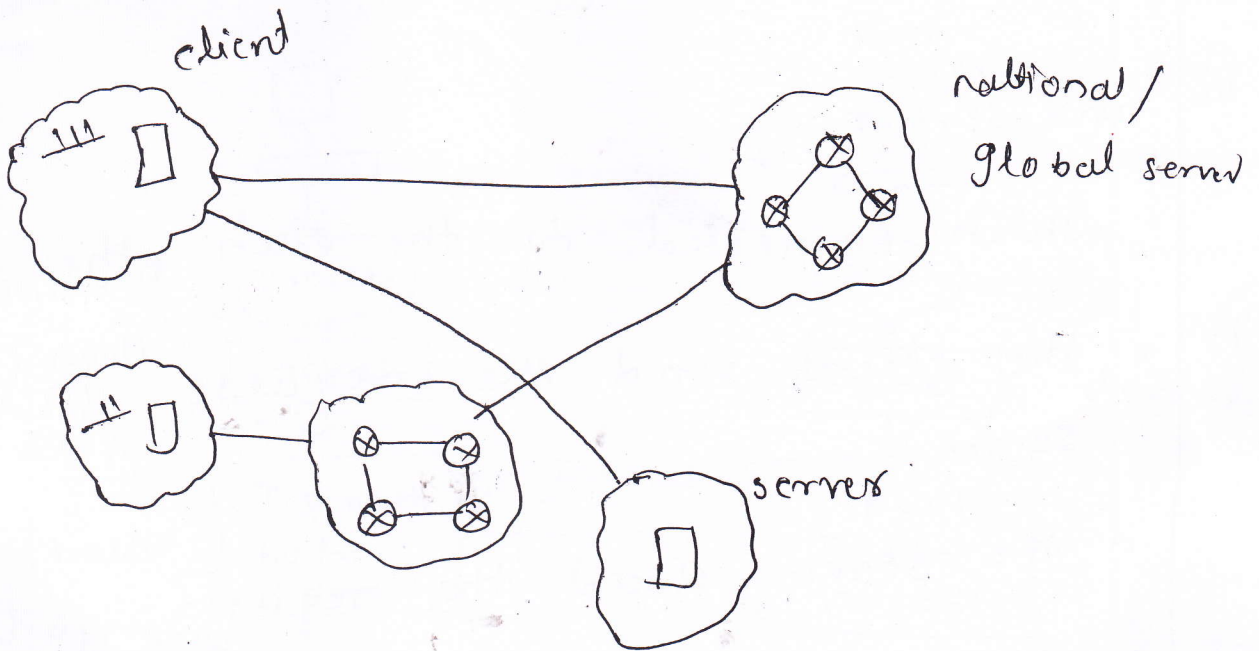


Application Layer:

1) a)

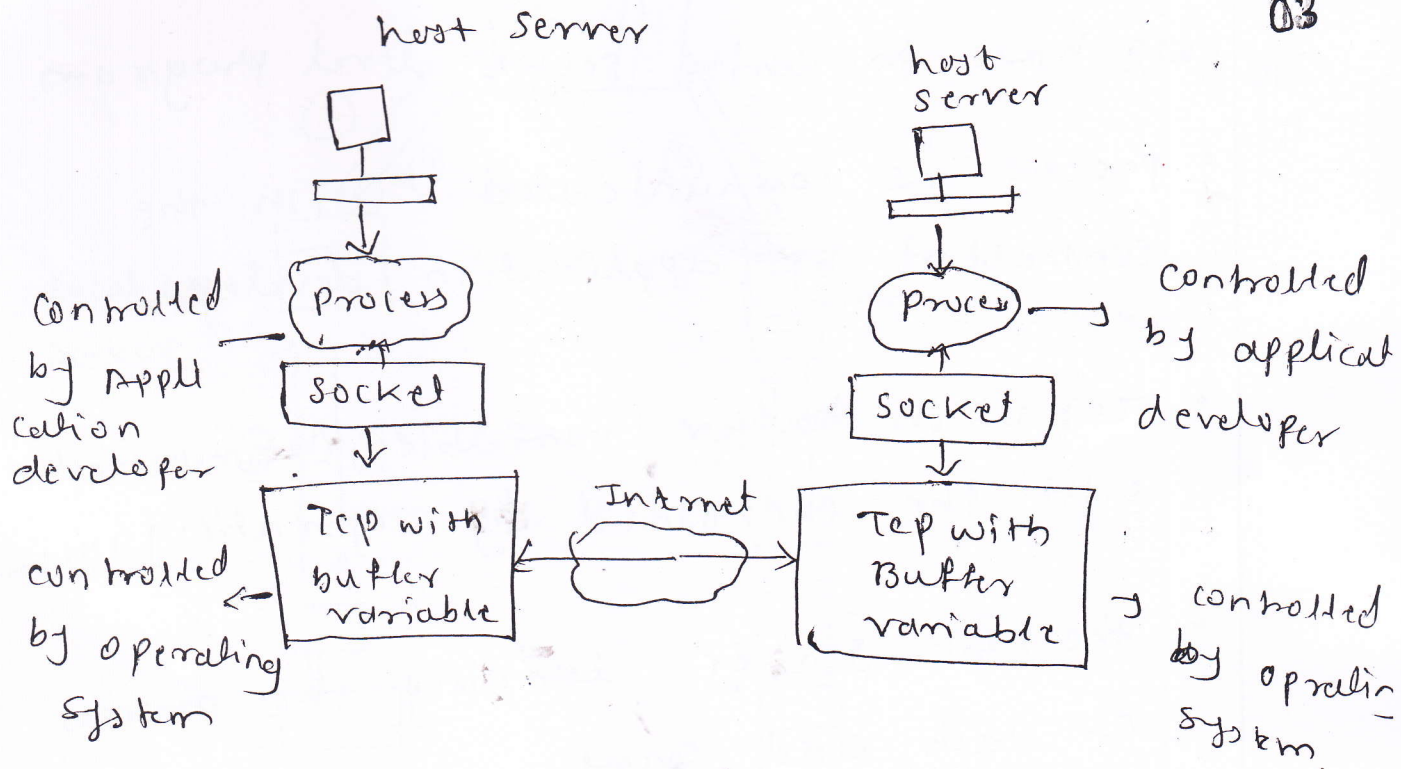


- as shown in the figure of client server architecture
- the client send the request to the server.
- the server read the message if it is exist
- then. the send the message to the client
- the three way handshaking is called
- the client & server is also called Round ~~Run~~ time.
- as show in the figure of client server architecture there are client, national global server

02

and server

- which connected to the each other
- the client send the request to the server.
- * the server Read the request.
- * if the request is exist in the server. then.
- * then the server response to client.
- * this is called client server architecture
- * the three way handshaking is also called
- * the client server architecture is also know as Round time transfer

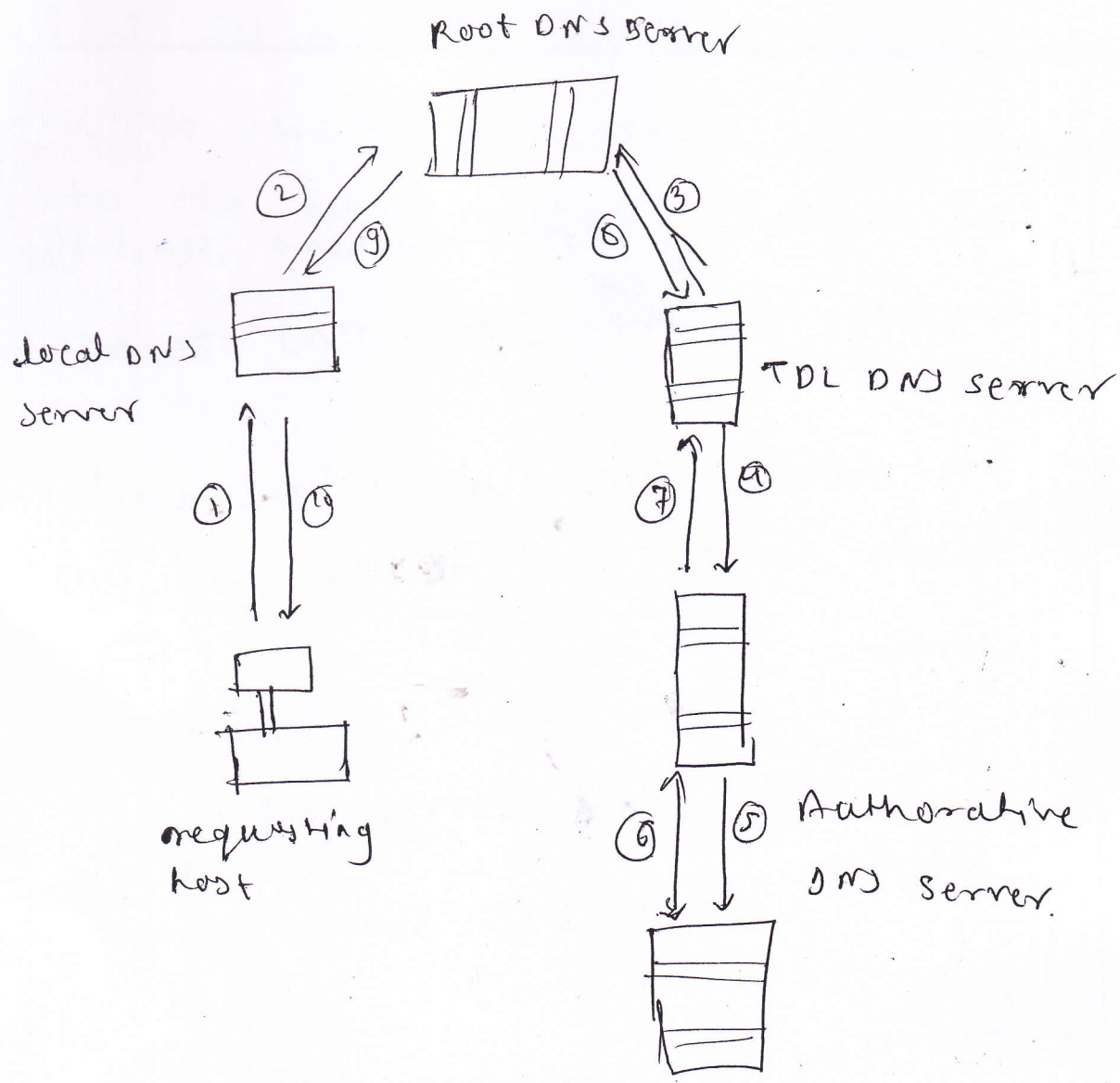


as show in the figure of Interface b/w the process and computer network

- there are host server, process, socket, tcp with buffer variable
- as show in the diagram of it is top down approach
- the relationship between the process connected with tcp with variable buffer
- the communication between the processor and tcp with buffer variable

04

- we have to write socket level program.
- process is communicated with the controlled by application developer
- Tcp with buffer variable communicates with the controlled by operating system,
- the host server looking the process of each and every working system
- the all are connect to each other
- host send the request to process and it will check the request messages
- and the process handle the control the application developer.
- and Tcp with buffer variable control the operating system.



as shown in the above diagram of Root DNS server.

there are many server that is local DNS server, Root DNS server, TDL, DNS server, Authoritative DNS server.

the the diagram there are many different server and only one request host machine.

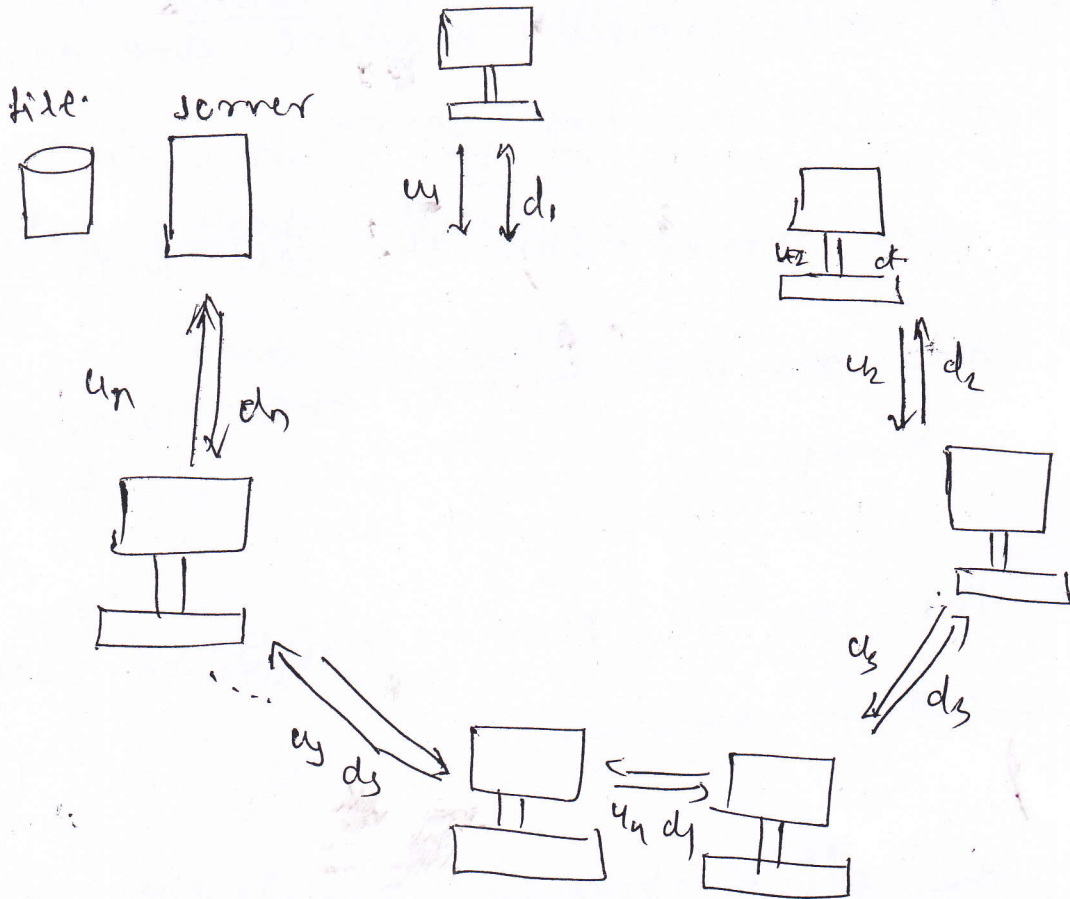
06

- the requesting host machine send the request to the local DNS server
- the local DNS server it read the data and send to the root DNS server
- the root DNS server sends it request to the TDL DNS Server .
- the TDL is nothing but ~~Top~~ Top Domain level.
- * the Top domain level is read the the message checking the data and read that data, send to the
- * last server that is authoritative server.
- * the authoritative server check strictly checking and response to ~~the~~ host requesting.

* this all are done by the servers.

* which checks the requests and response to the host machine.

u (b)



* as shown in the figure of P2P distribution
* the P2P is also called peer to peer distribution.

* the peer to peer is BitTorrent.

* the p

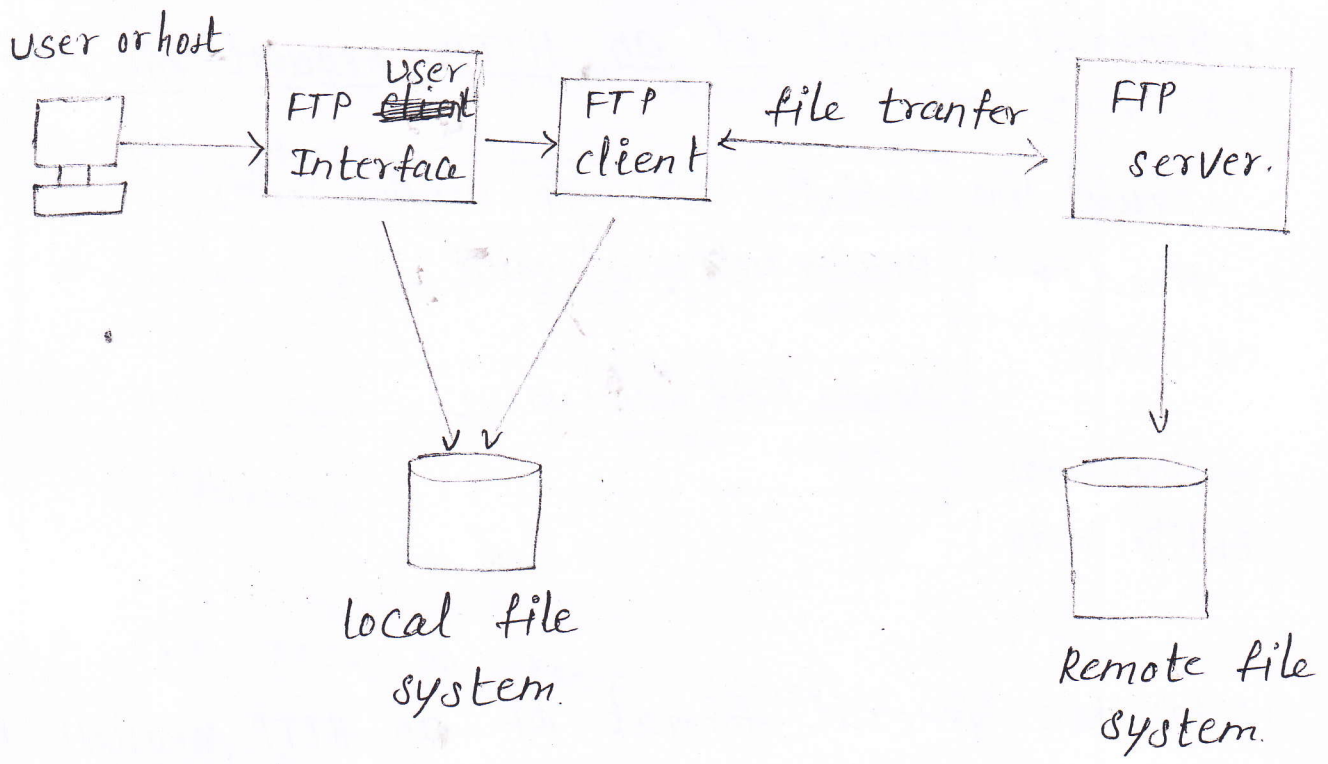
008

- * the peer to peer have the N that number of computer machine
- * and peer to peer have only one server
- * the all computer machine are connected to the one host server.
- * the server store the all information.
- * the ~~ser~~ peer to peer there is one disk storage
- * the peer to peer consists of one disk storage.
- * the disk storage consists of stores the data
- * the peer to peer ~~to~~ having the number of computer machine.
- * which is connected to the server.
- * the server checking and observing the all information of the ~~serve~~ computer machine.

• In this request message HTML version is used.

4] 5]

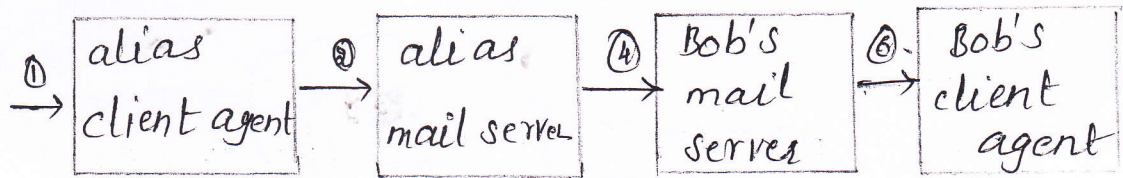
FTP (file transfer protocol).



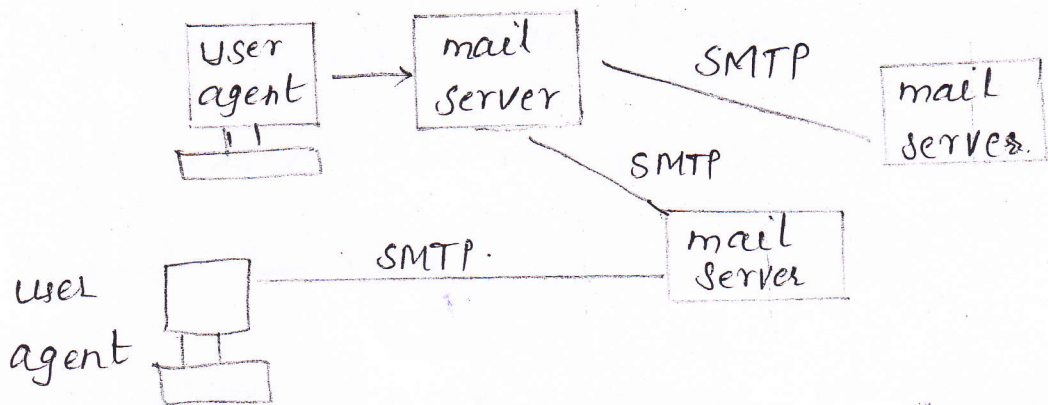
• In FTP user is sends the request to FTP user Interface then FTP client and FTP user Interface both joins and it create local file system.

• FTP client transfer files to FTP server and create Remote file system

- SMTP (Simple Mail Transfer protocol) :



- In simple mail transfer protocol alias client agent transfers the alias mail server.
- alias mail server transfers the bob's mail server and Bob's mail server is transfers to Bob's client agent.
- Electronic Mail :



- In this Electronic mail server SMTP is used.

Web server on TCP.

HTTP request msg initial connection established
different clients they are using in the server

client/server used the persistent HTTP; diff

Client & server exchange the message. &
response msg can be obtained after server
connection is established;

By request socket is created;

(i) connectionless Transport: UDP. (RFC 768)
x transport layer has to provide. mux or demux

socket level programming;

constant data query msg and passes msg to UDP
without performing handshaking performing hand
shaking

(ii) finer application-level control; UDP package the
data inside UDP segment and immediately
pass the segment to network layer
connection control mechanism

(iii) No connection established is 3 way handshake before
start to transfer data, data starts over UDP.
But FTP runs over TCP is critical,
But HTTP is used TCP rather than UDP

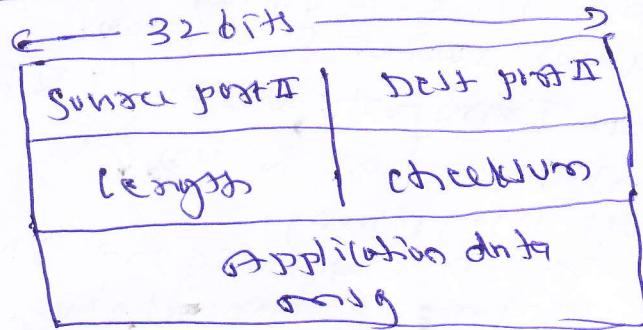
(iv) No connection state! TCP works based on the
connection but UDP is including connection state
and send buffers, congestion control, etc.
etc. But UDP doesn't having such type
of connection,
server supports for a particular applications
of UDP rather than many applications.

(v) small pkt header overhead! TCP segments has
20 bytes of header overhead in
every segment But UDP has only 8 bytes
of overhead.

UDP Segment Structure:

DNS server contains query msg or response msg
 UDP header has only 4 fields - each 2 bytes
 The length field specifies the number of bytes
 in the UDP segment, UDP provides the check
 sum if error is there or not,

UDP checksum



```

0 1 1 0 0 1 1 0 0 1 1 0 0 0 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
-----
1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 1
    
```

add sum of
~~two~~ two
 16 bit words

adding above 3rd word to the above sum gives

```

3rd word 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 1
          1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0
-----
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1
    
```

#00
 1's complement
 1's complement

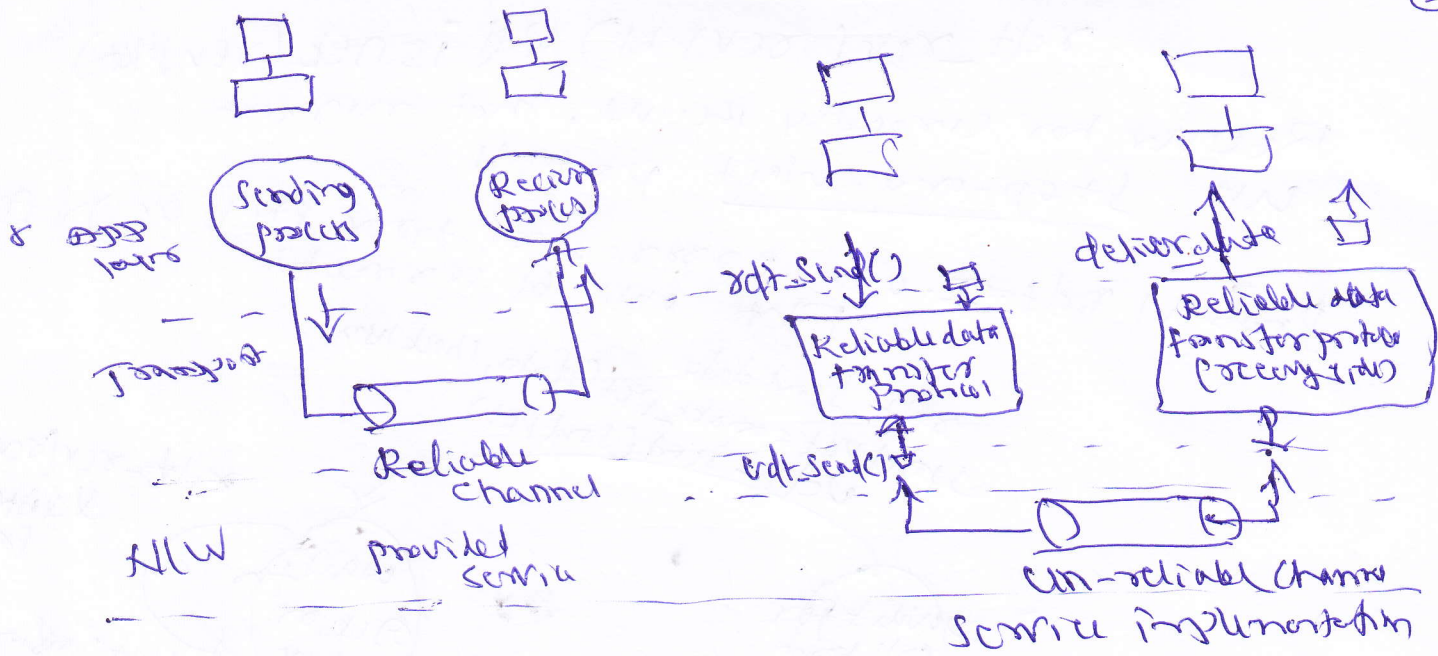
~~0000~~

```

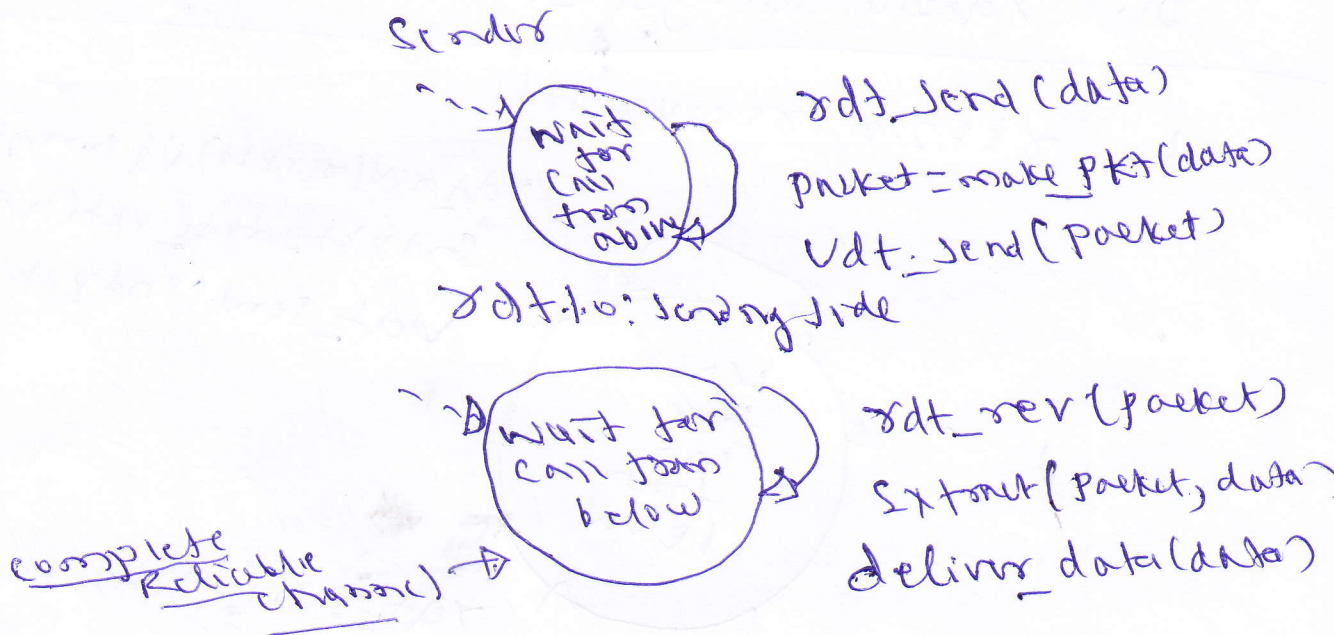
1 0 1 1 0 1 0 1 0 0 1 1 1 1 0
    
```

4 11 bits are added to checksum
 end to end principle.

- (14) principle of reliable data transfer:
1. occurs at transport layer & link layer
 TCP exploits many such a principle
 2. reliable channels also required



W) Building a Reliable data Transfer protocol
 Finite state machine, rdt 1.0 - y reliable channel



Reliable Data Transfer over a channel with bit error : rdt 2.0

+ve ACK - OK
 -ve ACK - Repeat

ARQ (Automatic Repeat request) protocol

Error detection: allow Rx to detect error
 require rdt 2.0

Receiver side: +ve ACK
 -ve ACK (Repeat)

Retransmission - pkt has received in error at the receiver