

Computer Networking.

Assignment-1

①. With the example explain the diff b/w HTTP persistent & non-persistent connections

→ HTTP persistence connection.

- * With persistence connection, the server leaves the TCP connection open after sending responses.
- * Hence, subsequent request & responses b/w same client & server can be sent over same connection
- * The server closes the connection only when the connection is not used for a certain amount of time.
- * Default mode of HTTP: Persistent connection with pipelining.

* Advantages.

1). This method requires only one RTT for all the referenced - objects.

or The performance is improved by 20%.

→ HTTP with non persistent Connection.

- A non persistent connection is closed after the server sends the requested - object to the client.
- * In other words the connection is used exactly once for one request & one response.
- * For downloading multiple objects, multiple connection must be used.

* Suppose user enters URL:

"http://www.someSchool.edu/some Department/home.
index"

Here how it works.

1. a) HTTP client initiates TCP

connection to HTTP server

(process) at

www.someschool.edu on port 80

1. b) HTTP server at host w

www.someschool.edu waiting
for TCP connection at port 80.

2. HTTP client sends HTTP

into TCP connection socket.

Message indicates that client
wants object.

Some Department/home.index

"accepts" connection, notifying client

3) HTTP server receives request
message, forms response message
containing requested object
& send message into its socket

5) HTTP client receives response
message contains .html file,

display html, Parsing html

files, finds 10 referenced jpeg objects

4) HTTP server closes TCP
connection

6) Steps 1-5 repeated for each
of 10 jpeg objects.

2) What is FTP? Explain the working principle of
protocol

* FTP is used by the local host to transfer files to or from a
remote-host over the network

* FTP use client-server architecture

* FTP uses 2 parallel TCP connections

① Control connection.

* The control connection is used for sending control
information b/w local & remote-hosts.

* Data connection

②

③ Data connection.

— The data-connection is used to transfer files.

Working.

1) When session starts the client initiates a control-connection with the server on port 21.

2) The client sends user-identity & password over the control-connection.

3) Then, the server initiates data-connection & then closes the data-connection.

4) Then the server initiates data-connection to the client on port 20.

5) FTP sends exactly one file over the data-connection & then closes the data-connection.

6) Usually the control-connection remains open throughout the duration of the user-session.

7) But the new data-connection is created for each file transferred within a session.

8) During the session, the server must maintain the state information about the user.

3. What is DNS? List its various services. Explain the following terms. (a) Host aliasing (b) mail server aliasing (c) load

+ DNS is an internet services that transfer translates domain names into IP addresses.

Services provided by DNS

The DNS is

1) A distributed database implemented in a hierarchy of DNS servers.

2) An application-layer protocol that allows host to query the distributed database.

- DNS servers are often UNIX machine running the BIND software.
- The DNS protocol runs over UDP & use port 53
- DNS is used by application-layer protocols such as HTTP, SMTP & FTP.
- Assume a browser request the URL www.some.edu/index.html.
- Next, the user's host must find obtain the IP address of www.some.school.edu.

(a) Host aliasing.

A host with a complicated hostname can have one or more alias names.

(b) Mail Server Aliasing.

For obvious reasons, it is highly desirable that e-mail addresses be mnemonic

(c) Load Distribution.

* DNS is also used to perform load distribution among replicated servers.

* Busy sites are replicated over multiple servers & each server runs on a different system.

411 With a neat diagram explain the structure of DNS Message format

→ 2 types of message

1) query 2) reply

Both query & reply messages have the same format

③

Identification	Flags.
Number of questions.	Number of answer RRs
Number of authority RRs	Number of additional RRs
Questions (Variable number of questions)	
Answer. (variable number of resources.)	
Authority. (variable number of resource records)	
Additional information. (variable number of resource records)	

• Various field in DNS message are as follows

1) Header Section.

- The first 12 Bytes is the header-section.

This section have

(i) Identification (ii) Flag. (iii) Four Number of Fields

② Question Section.

- This section contains information about the query that is being made.

- This section has following fields:

(i) Name (ii) Type.

③ Answer Section

- This section contains a reply from a DNS server.

- Contains the resource-records for the name that was originally queried

④ Authority Section.

- contains the record of other authoritative-server

⑤ Additional section.

This section contains other helpful records.

5. Explain the working of multiplexing & de-multiplexing.

- * A process can have one or more sockets.

- * The sockets are used to pass data from the network to the process & vice versa.

→ Multiplexing.

- * Multiplexing is done at the sender end in the transport layer.

- * Gather the data-chunks at the source-host from different sockets.

- * Encapsulates data-chunk with headers to create segments &.

- * Passes the segments to the network-layer.

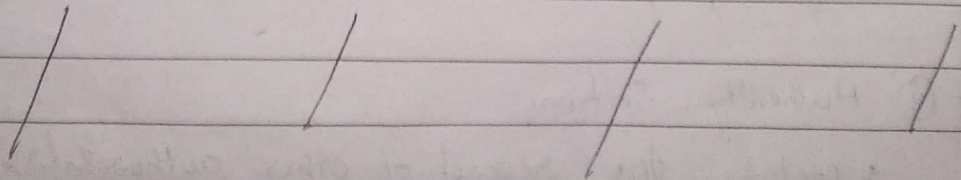
The job of combining the data-chunks from diff. sockets to create a segment is called multiplexing.

→ Demultiplexing.

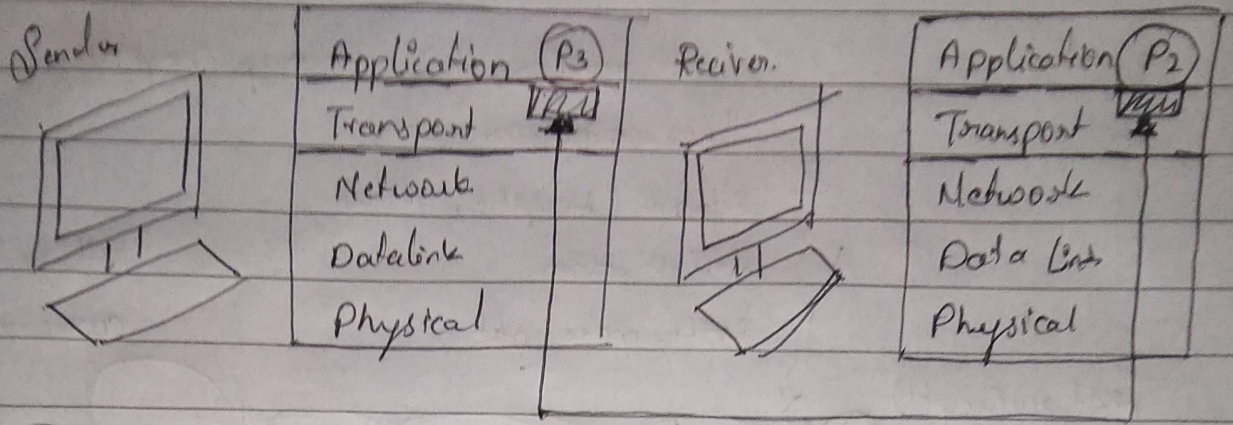
This is done at the transport layer of the receiver end.

- * Examines the fields in the segments to identify the receiving-sockets &.

- * Directs the segments to the receiving socket. The job of delivering the data in the segments to the correct socket is called de-multiplexing.

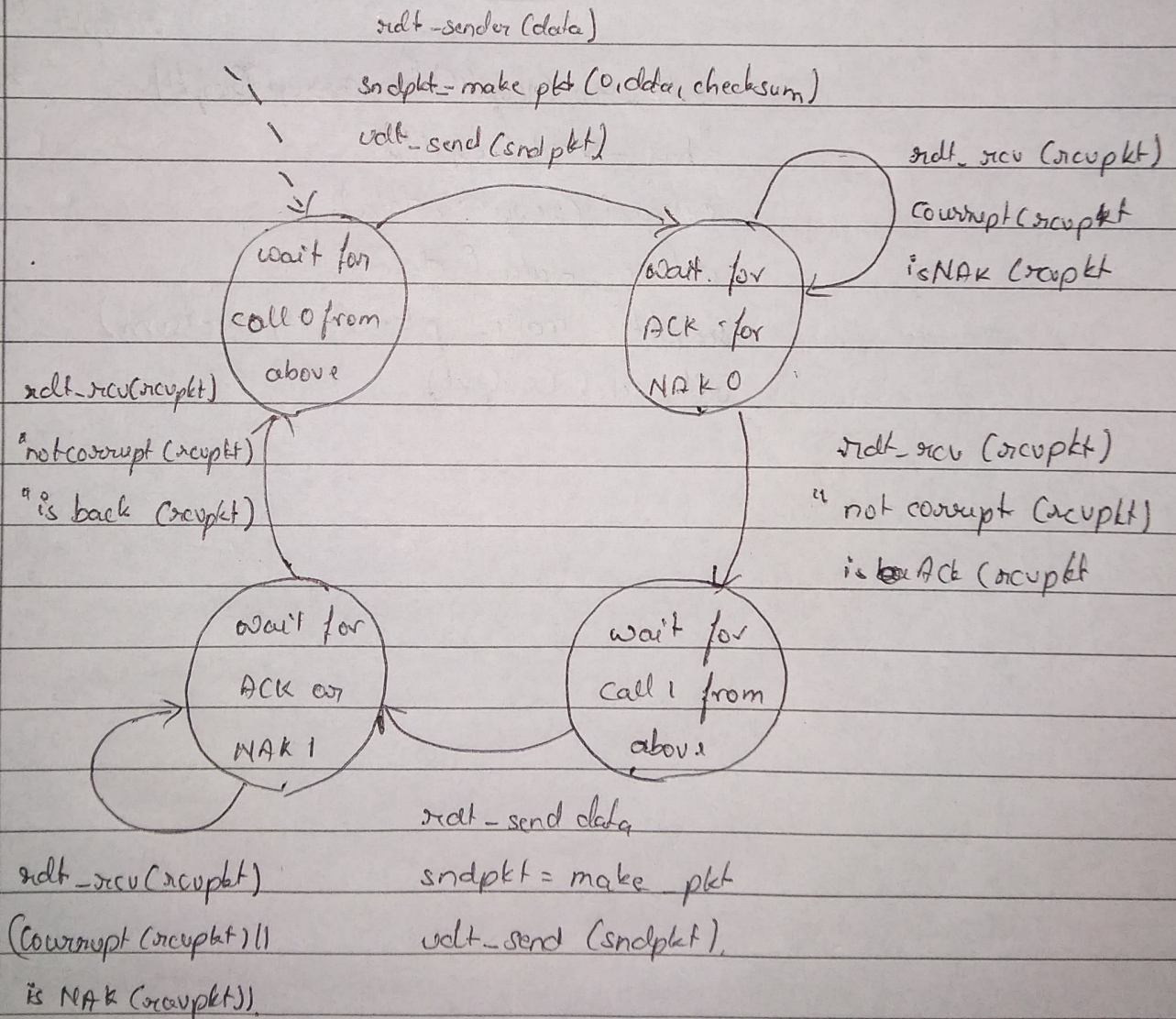


④



○ process TCP Socket.

⑦. Explain the working of Rdt 2.1 using diagram.



rdt - 2.1 sender.

rdt-rev (rcv pkt) "not corrupt (rcv pkt)"

"has - seq (rcv pkt)"

extract (rcv pkt, data)

deliver_data.

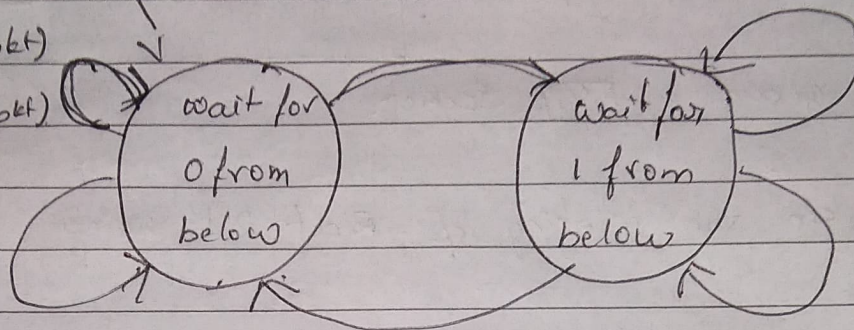
sndpkt = make_pkt

rdt-rev (rcv pkt)

corrupt (rcv pkt)

rdt-rev (rcv pkt)

corrupt (rcv pkt)



rdt-rev (rcv pkt)

not corrupt.

rcv pkt "has seq 0."

has s - eq 0."

rdt-rev (rcv pkt) "non corrupt (rcv pkt)"

"has - seq 1 (rcv pkt)"

extract (rcv pkt, data)

deliver_data (data).

sndpkt = make_pkt (ACK, checksum)

rdt-send (sndpkt)