भारतीय प्रौद्योगिकी संस्थान खड़गपुर

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Department of Computer Science and Engineering

CS 31006: Computer Networks | **Mid Semester Examination**

Full Marks: 50 | Time: 2 hours | Spring, 2019

Note:

(i) There are THREE questions in this paper. Answer all the questions. The answers should be precise and to-the-point.

(ii) Write down the assumptions clearly, if any. No queries will be entertained during the exam hours.

4 1. (a) Assume that you are within the IITKGP network, and your name server IP is 202.141.80.81. You want to access a website `www.mynetworkcourse.mooccourses.ac.jp`, which is a newly launched website, and the address information has not been cached in any of the intermediate DNS server. The `ac.jp` name-server maintains the authoritative resource record for *this website*. Explain with a diagram, how the name resolution will be done when you throw a *DNS query for this website name*. [4]

4 (b) What are the purposes of connection establishment at the transport layer? Can the transport layer ensure reliability without establishing a connection? [2+2]

4 (c) Assume that you want to run two different instances of HTTP servers, which will host two different sets of contents. What can be the problem if you run them on the same port address? Assuming that the two servers are hosted on ports 80 and 8080, respectively, and the domain name of your machine is `www.mypages.ac.in`; what will be the two URLs to access the two servers? Consider that you want to access the `index.html` web-page which is hosted in both the servers. [2+2]

3 (d) Consider the following HTML page `mypage.html`.

```
<!DOCTYPE html>
<html>
<head>
    This is a sample page for the CS31006 course
</head>
<body>
    <img src="network.gif" alt="HTML5 Icon" width="128">
    <img src="images/cs31006.gif" alt="HTML5 Icon" width="128">
</body>
</html>
```

How many HTTP GET requests are required to render this page properly on a web browser. Write down those GET requests, assuming that the domain name is `www.mynetworkcourse.org` and you are using HTTP version 1.1. A sample GET request looks like as follows.

Scanned by CamScanner

```
GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1
```

[3]

3 2. (a) Why FTP uses a separate control connection to transfer the FTP commands rather than using a single connection for transferring both the commands and the data? [3]

4 (b) Assume that you initiated a transport layer connection with an initial sequence number 3200 that follows byte sequence number over a 12 bit sequence number space. You have generated 5 data segments, each of size 200 bytes over that connection, and then the connection got crashed. Consider that the lifetime of a segment over the network is 5 seconds, and at every second, you can generate 100 bytes with unique sequence numbers.

After the connection got crashed, say, you have initiated a new connection after 3 seconds from the same port to the same server with an initial sequence number 100. Do you think that this new connection is safe? – Explain with clear justification. [4]

4 (c) Why can't we use SMTP to retrieve emails from a mail transfer agent (MTA)? I design a modification over SMTP as follows: *The MTA will poll for the user agents (UAs); whenever it will find out that a UA is online, it will use SMTP to push the email to the client mailbox at the UA.* Do you see any problem in this modification of the protocol? [2+2]

4 (d) During the release of a transport layer connection, why a timer with a predefined timeout is started after sending the connection release message? Does this procedure ensure that no in-flight data will be lost during the transmission? [2+2]

5 3. (a) Consider a sliding window based ARQ with selective acknowledgments to acknowledge out-of-order received segments, where the sender window size is M, and receiver window size is N. The protocol uses a p bit sequence number space. What will be the maximum value of M such that a sequence number wrap-around does not create problem in distinguishing between a fresh segment and a re-transmitted segment? Give justification for your answer. Consider that both N and p are fixed and cannot be tuned. [5]

6 (b) Consider a Go-Back-N based sliding window flow control algorithm (receiver window size is 1) running over a 1 Gbps link. The end-to-end delay between the sender and the receiver is 20ms. What should be the minimum window size in bytes (express in power of 2) to ensure maximum link utilization under this environment? Considering byte sequence numbers, what should be the maximum number of bits in the sequence number space to ensure correct operation of the protocol? [3+3]

3 (c) In sliding window based flow control algorithms, the sender window size is kept no more than the receiver advertised window size. Explain, how this can lead to a potential deadlock situation. How does the algorithm need to be tuned to come out of the deadlock? [2+1]

3 (d) The general principal of congestion control at the transport layer is to detect congestion once it happens; and then response to it to come out of the congestion. This is a restrictive approach. Explain, why can't we simply avoid the congestion (ensure that congestion does not happen at all) by controlling the sending rate from the transport layer? [3]

3 (e) Why do we prefer adjusting the sender window based on an additive increase multiplicative decrease algorithm, for ensuring congestion control in the network? Explain with an example. [3]

Scanned by CamScanner

2a) The FTP → File Transfer Protocol is used for transfer of large files over a TCP connection.

As mentioned, the time taken for these large chunks of data to be transfered is also very high. Therefore, the data sending channel remains visibly occupied for a decent amount of time.

Now, if the command channel is the same as the data channel, during this duration of data transfer their can be no exchange of commands.

3

Neither can the Server cater to other clients thereby increasing their queing delay, the current client's commands of STATUS, ABORT, QUIT can not be handled.

Hence in order to be able to cater to more than one client parallely and process commands whilst a large data transfer takes place.

FTP uses two channels.

Scanned by CamScanner

(b) 12 bit Seq Num $2^{12} = 4096$.

Seg 1	3201	3200
Seg 2	3401	3600
Seg 3	3601	3800
Seg 4	3801 . . .	4000
Seg 5	4001	4200 (104)

Wrap Around

Hence the last segment, Seg 5 contain the Seq Numbers.

$$4001 \ldots\ldots 4095 \ldots 0 \underbrace{\ldots}_{100.} 104$$

So the last Sequence has 100 as one of the Sequence Numbers.

Since the crash takes place after 5 segments, it can be assumed crash happens almost as soon as $\boxed{\text{Seq-Num } 100}$ byte was generated.

Now, this particular Packet would live for 5 secs but we initiate our connection after 3 secs, when this packet is alive, with a $\boxed{\text{Seq_Num} = 100}$ same as this packet.

Scanned by CamScanner

Thus, in our Network we have two packets of same seq-num but not duplicates of each other.

4 This may result in us falling for the delayed duplicate conundrum → NOT A SAFE Connection

(G). There are certain reasons why we cannot use SMTP to retrieve mails as well.

(i) SMTP is fundamentally a push-based protocol not a pull-based.

2 Retrieve Emails ⇒ pulling

(ii). The Receiver UA might not be active all the time, which is why it is tedious to use SMTP to do the final delivery as well. Better to leave it at the UA's discretion.

This modification employs SMTP's push to further extend till the Receiving UA when it is online, as the MTA poll results would indicate.

2 Lots of intricacies to think about here.

⊙ Frequency of these polls, – How often will the MTA conduct these polls, there might be a case when the UA is online but no poll yet so UA has to wait for its mails at the Mercy of the MTA-poll.

Scanned by CamScanner

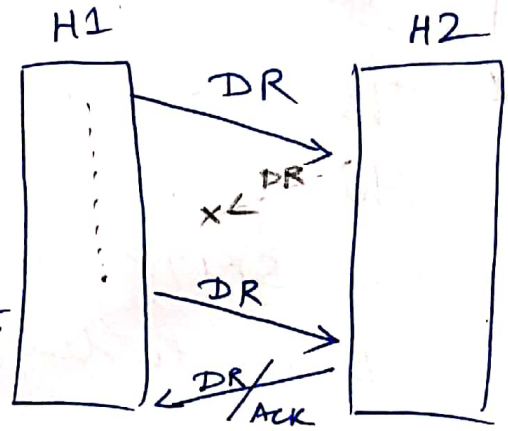
• <u>Solution</u> - we may use some sort of triggers which would tell us as soon as some UA is online.

May have high implementation overheads

• A foreign agent would want to connect to the (receiver's VA) system and wish to transfer data. Firewall issues might exhibit

Hence there are definitely lots of problems in automating this process through SMTP, its best to leave it at the VA's PULL Mercy.

(d). The release connection is symmetric process where both systems wait for an ACK for their DR request before actually terminating the connection.



As shown in the Figure, in case the ACK(DR) by Host 2 is lost, the H1 will keep waiting and keep on doing so.

2

Here the Timer is important as for a long time inactivity, Host 1 timesout and sends back the DR in anticipation for an ACK(DR) from Host 2 this time around.

Scanned by CamScanner

The predefined time is typically the average expected/set Lifetime of a packet in the network if this much time elapses, we can assume packet to be dead.

The Two Army Problem clearly tells us that it is not possible to have a perfectly synchronised release of two connections. However this mechanism we adapt reduces any asynchronisation or in-flight data loss as much as possible.

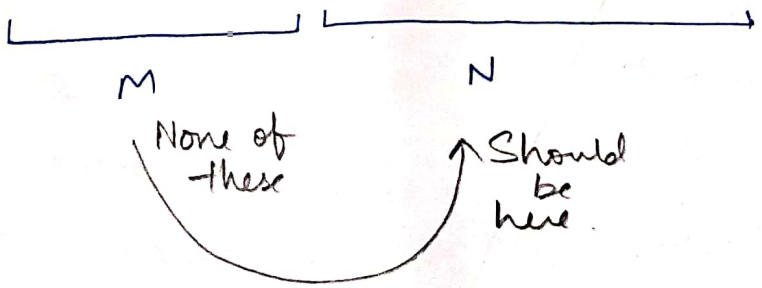
If their is some data inflight yet to be acknowledged which was sent by Host 2, it will not ACK(DR) to Host 1's request and REJ its request to release connection. This way Data Release occurs only when both sides are in consent.

2 So in-flight data loss is taken care of.

Scanned by CamScanner

3a) Sender Window Size = M
Receiver Window Size = N
p-bit Sequence Number Space

Say Sender (S) sends M packets to the Receiver (R), which get Received, However their ACKs get lost.
As a result.
R's window moves ahead by a distance M but Sender's Window has all packets Outstanding (C).
So Sender shall resend these packets.
Typically Receiver should recognise these packets as duplicate and discard them, while resending the ACKs.
For this reason, none of these M seq nums should be in the new R' receive window range



Hence; ATLEAST (M+N) diff seq nums

2

Scanned by CamScanner

Thus $2^P \geq M + N$.

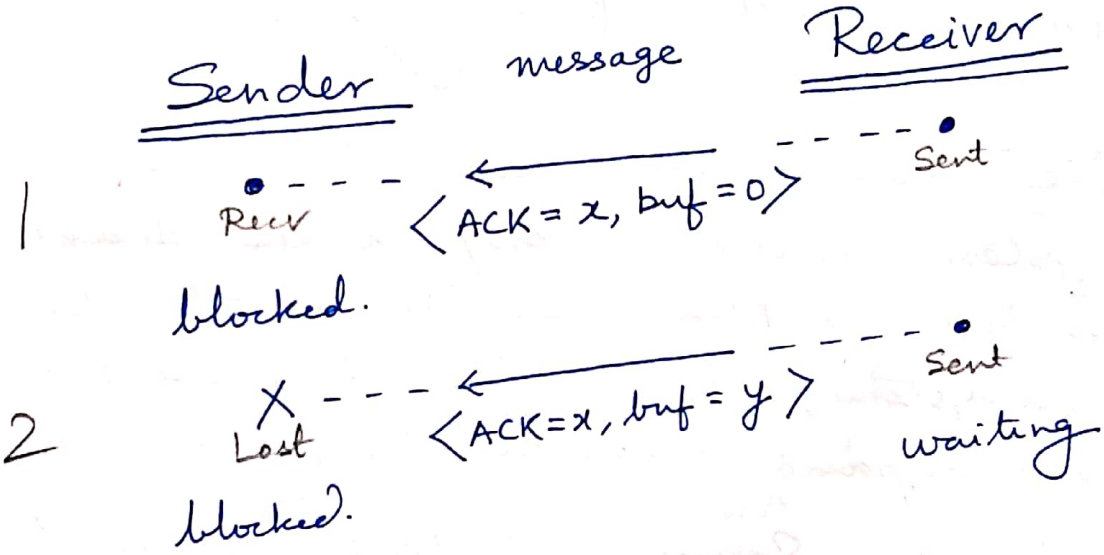
$$\boxed{M \leq 2^P - N}$$

Max value possible for M $\rightarrow (2^P - N)$.

5

Good

(c).



In {message, Command, Request} 1, the Receiver allocates a buffer space of 0 to the Sender effectively blocking it. It then send 2nd Req with same ACK allocating it space $y > 0$, which gets LOST. So here's the situation.

2

S → Blocked since buffer space allocated → 0

R → Waiting for sender to send as it has been allocated space

This is thy typical situation of a Deadlock

Scanned by CamScanner

The Receiver side in general is not programmed made that way where it can handle such a situation.

However what we can do here is, we can make the Receiver timeout if it has been waiting inactive on a connection that is still open. Simply Resending the previous ACK after a timeout will suffice to solve this deadlock.

(d). The Network of server-client-switches systems is a very complex and decentralised graph where Nodes are the Switches Routers and End Systems, whereas the edges are the links and edge capacities can be thought ~ bW capacity for a given Source-Sink Pair., we can find out the optimal rate of sending by the Max Flow Min Cut Algorithm.

However this does not work.

(1) Algorithms like these can only hold fast for static networks which have only one Source Sink.

Moreover this Algorithm needs information of the whole network to be effectively processed and even then transmitting that processed data through the whole network so that each node performs its correct task is a far stretch.

Scanned by CamScanner

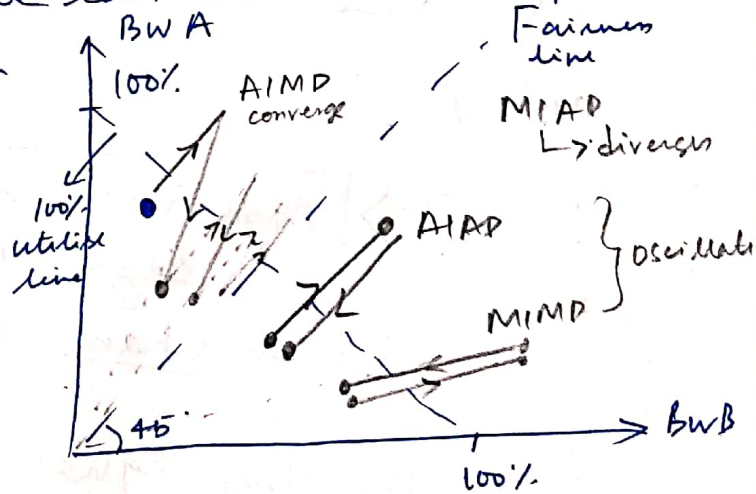
(ii) There are ALMOST NEVER only one Source Sink pairs in a Network.

Flows appear and terminate dynamically and since we have no A-Priori knowledge of this implementing an Optimal Flow Algorithm Dynamically is not possible.

Hence we cannot AVOID congestion. All we can do is, simply detect and take action against it.

(e) There are two major reasons why we choose AIMD for the control congestion mechanism

(1). <u>optimality</u>. An Additive Increase (+45°) and a Multiplicative Decrease (towards origin) is the only way we can achieve the optimal intersection point of 100% link utilisation as well 100% fairness → 50% - 50% B'width to both the flows.



3

(ii) <u>Stability</u> - it is easier to fall into congestion than to recover from it. Hence a slower increase (Add) and a faster (rapid) decrease (Multiply) is also favoured for Stability Reasons

Scanned by CamScanner

(b) 1 Gbps link speed.
20 ms delay (end to end).
40 ms (RTT).

Twice Bandwidth Delay Product $2 \times \boxed{BDP}$ $= 2^{30} \times 40 \times 10^{-3} = 2^{32} \times 10^{-2}$

$2^{32} \times 10^{-2}$ bits.

$= 2^{29} \times 10^{-2}$ bytes

$= 1.28 \times 2^{22}$ bytes

3

(Min Window Size.) $= 1.28 \times 2^{22}$ bytes.

$\boxed{\text{Max outstanding Segments in Channel}}$

By this equality chain, we get

$W = 1.28 \times 2^{22}$ bytes

Scanned by CamScanner

Number of bits $= p$.

$(2^p) =$ Window Size

$p = \log_2(\text{Window Size})$.

$= \log_2(1.28) + \log_2(2^{22})$

$p = 22 - 6$

$= \log_2(2.56) + \log_2(21)$

$= 21 + 4$.

Hence; $\boxed{21 < p < 22}$

3

Therefore.

$\boxed{\text{bits in seq num} = 21.}$

Scanned by CamScanner

1)
(b). Connection establishment at the transport layer is important because there is a need to maintain certain state variables of the connection for correct functioning of the server-client connection system. (c)

Good

2 The major reason of connection establishment is in fact Reliability. 2

In Reliable Data Transfer, what we do is we perform checks after almost each step to ensure the successful delivery and acknowledgement of data at the other end. This Checking acknowledg paradigm is what forms the Reliable Wrapper around the unreliable Network Layer connections 2

2 Maintaining information of state of sender/rec next sending seq num, last Acked seq num etc is fundamental to Reliability, and this cannot be done without Connection Establishment

Good

Scanned by CamScanner

(c) If we try to run both instances of the same application on the same port, we will not be able to get the "different set of contents" which we desire.

2 This is because Applications in the Kernel are identified by their Port Numbers.
Using same port number for two different instance would simply lead to faulty undesired functionality

The two different URLs shall be

http://www.mypages.ac.in :: 80

2 http://www.mypages.ac.in :: 8080.

to distinguish between the two ports.

Scanned by CamScanner

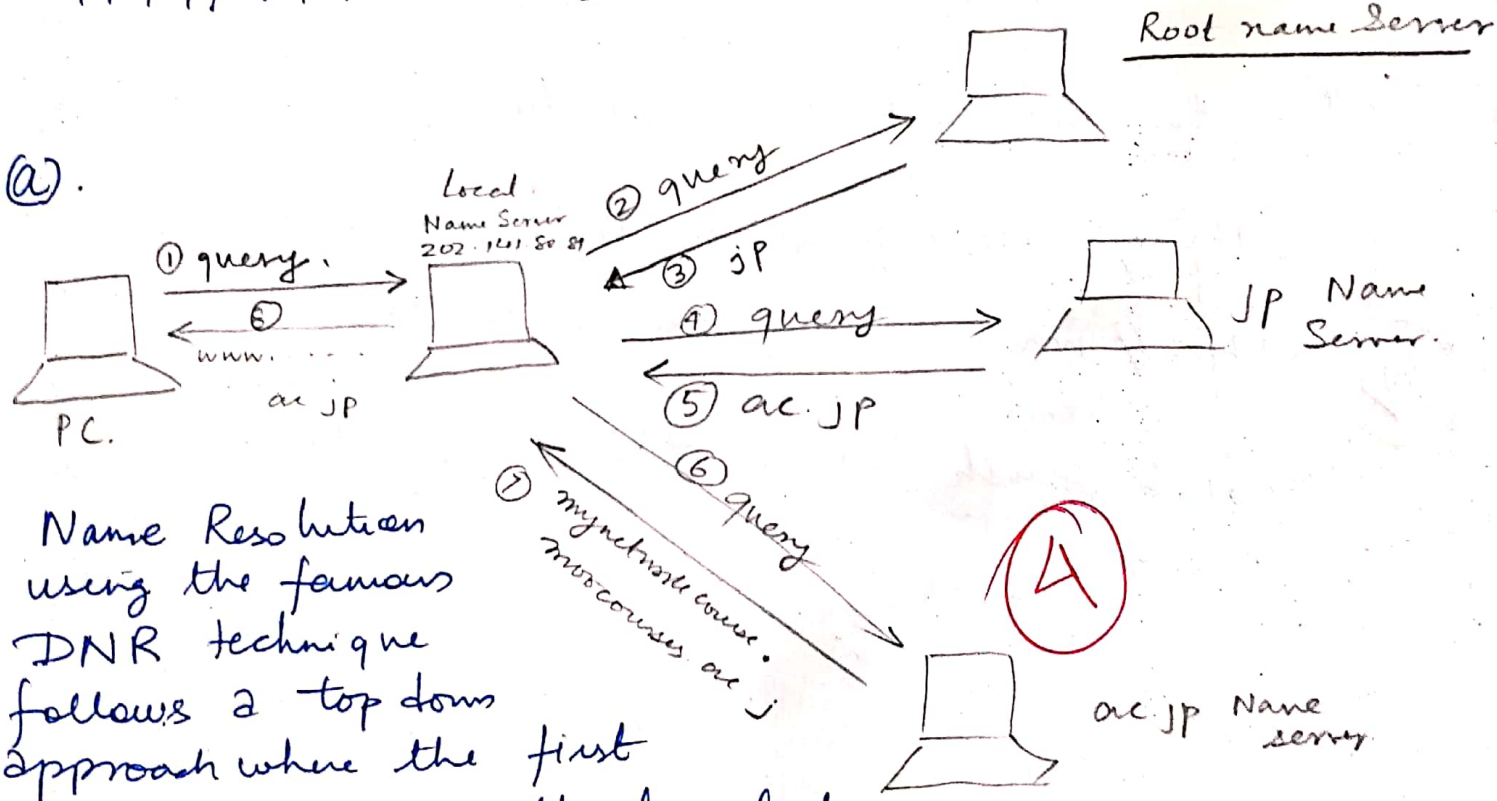
(d)

3

GET http://www.mynetworkcourse.org/mypage.html HTTP/1.1
GET http://www.mynetworkcourse.org/network.gif HTTP/1.1
GET http://www.mynetworkcourse.org/images/cs31006.gif HTTP/1.1

These are the 3 get requests that will be required to render this page properly on a web browser.
HTTP: 1.1 → may also allow pipelining.

(a).



Name Resolution using the famous DNR technique follows a top down approach where the first domains detected are the broadest ones, and then we continue to narrow down on the correct domain address.
The Local Name Server Recursively keeps on querying the different Root Servers/Name servers till the final answer is received from any of them. The other servers also iteratively keep hearing to clients and resolving their queries (parts of it). The process terminates when we get an Authorative Record and then the Record is sent to User's PC by Local Name Server

Scanned by CamScanner

Full Marks: 25 ...puter Networks

Time: 45 Minutes

Class Test – 1
Spring, 2019

1. State whether the following statements are TRUE or FALSE. Write a justification (within maximum 50 words) against your answer. **Note: Marks will only be given if both the TRUE/FALSE assignment as well as the justification are correct – there is no separate marks for justification only.**

(a) Any client-server based protocol at the application layer must maintain the states of the clients at the server. [NB: Maintaining states means remembering the sequence of messages that have been transferred from the client to the server] [2x5]

FALSE – As we know HTTP is by defaut stateless. And it works fine enough, so there is no necessity to maintain states. Moreover, the memory that would be required to save this info on the server side to the huge number of clients would be huge, hence not feasible, not really nd.

(b) Multiple GET requests can be pipelines in HTTP1.1.

False – Since URLs are cascaded in the HTTP Responses, it may be that on pipelining a cascaded query's response reaches us before the enclosing page's query's response, which would render a difficult error situation to HANDLE.

... used to uniquely identify two different connections from the server

Scanned by CamScanner

(b) Multiple GET requests can be pipelines in HTTP1.1.

False - Since URLs are cascaded in the HTTP Responses, it may be that on pipelining a cascaded query's response reaches us before the enclosing page's query's response, which would render a difficult error situation to HANDLE.

(c) Port addresses can always be used to uniquely identify two different connections from the server.

FALSE- On the server side, for all the Standard Protocols, the port numbers are fixed. However still if we receive the same port number in a new connection with one already there, we do not know if this is a new connection, or the server crashed and wants to re-establish the old connection.

2

Scanned by CamScanner

(d) Any two Internet Service Providers (ISPs) in the Internet can always directly exchange data between themselves, without involving any third-party ISP.

FALSE → Physical Constraints like distance and availability may hinder two ISPs from establishing a direct connection always.
Hence, not always is this TRUE so sometimes there is a need of a third-party ISP.

(e) No timeout is required in case of Stop-and-Wait ARQ.

FALSE - Stop and Wait ARQ actually needs to wait for Acknowledgement of the previously sent data packet. In case it does not arrive within a certain time interval, a "TIMEOUT" occurs and the packet is resend instead of waiting for the ACK.

Scanned by CamScanner

2. Answer the following questions. The answers should be precise and to the point. Unnecessary write-ups may result in penalties in the form af additional marks deduction.

(a) Why e-mail uses two different set of protocols (SMTP and IMAP/POP) for email exchanges? Clearly mention the purpose of each set of protocols. [3]

The SMTP protocol's task is to deliver the message from the machine of the sender to the server local on the receiver's side.

The POP3 / IMAP help the receiver to "pull" the message from its local server to its disk

This is done because of the fact that if there was just one protocol which handled the complete route, then the presence/availability of both the sender and receiver needs to be required for the successful transmission of the message.

The way it is done now enables the sender to be releived of their duties after the send to the receiver's local server is accomplished and then the receiver can arrange a pull of the unread messages at their leisure.

3

Scanned by CamScanner

(b) Why UDP is a preferred choice for DNS query and response messages? [3]

- Speed - UDP is faster than TCP in the sense that there are no HANDSHAKING OVERHEADS. Since there is a cascade of requests, UDP at each connection would save a lot of time as compared to a TCP with handshakes at each of them.

- The responses are small enough to fit one UDP segment hence no synchronisation, FIFO type problem occurs in this case.

- Reliability - Though UDP is unreliable as compared to TCP, the Reliability here is ensured by the wrapping application, which keeps on sending requests if it encounters timeouts till a user defined N times before reporting a FAILED CONNECT error me

3

(c) Why HTTP is not suitable for very large file transfer between the server and the client? [3]

Scanned by CamScanner

(c) Why HTTP is not suitable for very large file transfer between the server and the client? [3]

The HTTP protocol is essentially developed for a HyperText Transfer, which involves certain HTML encoding with URLs to be transferred. Jamming the connection with a Large File Transfer may lead to Queing delays in other subsequent responses, hence degrading the Quality of Service.

Disadvantage of single connection channel hence, we prefer FTP for this purpose.

2

Scanned by CamScanner

(d) Why do we implement flow control both at the transport layer as well as at the data link layer? [3]

Flow control at the Transport Layer is essential because since the end-to-end picture is known at the Transport Layer, we can choose an optimal transmission rate after observing the different link speeds present in the ROUTE. This ensures that their is MINIMAL QUEVING DELAY and close to Nil chances of Packet Loss occurring during the Transmission.

At the Data-Link Layer, the situation is more local, concentrating on particular Hubs. We need to decide how much of data inflow we can allow to be able to flow-out this data without delay and loss. This improves the performance of the transmission.

Since same links can be used for different connections the Hub Level data flow controls that there is minimal delay and NO loss.

2.5

(e) For the connection establishment at the transport layer, why don't we use the immediate next

Scanned by CamScanner

(e) For the connection establishment at the transport layer, why don't we use the immediate next sequence number from the one that the last connection over the same port had used? [3]

The Delayed Duplicate problem is one in which when we receive a connection with same sequence number as one already alive, we do not know whether it is a new connection or a duplicate to the alive one. Each connection starts with an initial sequence number and there is an integral increase in the sequence numbers used in that connection since each sequence number is associated with only one byte. Hence there are high chances that the next sequence number to the initial one which initiated the connection is in use and may be alive as well. We can however wait for lifetime (T) of a sequence number/packet after which any sequence number can be used to initiate a connection, however with some waiting time loss.

3

Scanned by CamScanner