Department of Computer Science and Information Systems, BITS Pilani
Second Semester 2018-2019
Computer Networks (CS F303)
MID SEMESTER TEST Regular (CLOSE BOOK) SOLUTIONS

Duration: 90 Mints (9:00 to 10:30 AM) **Date: 16-MAR-2019** **MM: 40**

Note: All questions are compulsory. Answer sub parts of a question (if any) together.
Assume 1K = 10^3, 1M = 10^6, 1G = 10^9

Q.1 Using a Web browser, you visit the web site for www.somesite.com. The base HTML page for the main page is of 35,000 Bytes. Once the base HTML page is fetched, it contains URL references for the following embedded images: (Note: Image size is specified in parenthesis.) [2+2+2+2 = 8M]

http://www. somesite.com/banner.jpg (15,000 Bytes)
http://www. somesite.com/logo.jpg (5,000 Bytes)
http://www. somesite.com/picture1.jpg (10,000 Bytes)
http://www. somesite.com/picture2.jpg (10,000 Bytes)
http://www. somesite.com/picture3.jpg (5,000 Bytes)
http://www. somesite.com/picture4.jpg (35,000 Bytes)

Take the following assumptions:

- At most 10,000 Bytes of data fits into a single packet. Ignore the overhead of any headers.
- HTTP requests are 1,000 Bytes in size.
- Ignore the time required for closing connections and the delay introduced in acknowledging the final data packet sent by the server to your browser.
- All TCP senders use window size of 20,000 Bytes.
- No packets are lost.
- The IP address has been resolved by DNS query.

a) For the initial transfer of the home page, how many RTTs are required, and what occurs during each of them?

Sol: 3 RTTs required.
1 RTT for the TCP connection establishment.
1 RTT for sending the request for 20,000 Bytes and receive the main page data in reply.
1 RTT for sending ACK and receiving remaining 15000 Bytes of data of main page.

How quickly (in terms of RTTs) can your browser download all embedded images of the web page if the browser uses: (Note: Give appropriate reasons in support of your answer.)

Note: Answers for b), c) and d) can omit 3RTTs required for fetching main page.

b) One connection per image, with up to 4 concurrent connections.

Sol: Total 8 RTTs (3 RTTs to fetch the main page + 2 RTTs to fetch first four images + 3 RTTs to fetch last two images)
OR
Total 7 RTTs (3 RTTs to fetch the main page + 3 RTTs to fetch picture4 and any three other images + 2 RTT for remaining two images)
Note: 1 RTT for picture4 and remaining two images is overlapping.

c) A single persistent, non-pipelined connection.

Sol: 10 RTTs required
3 RTTs to fetch the main page
5 RTTs (1 RTT for each image other than picture4)
2 RTTs for picture4 (as window size is 20,000 Bytes)

d) A single pipelined connection in which server can respond to requests in asynchronous manner.

Sol: 7 RTTs
3 RTTs to fetch the main page.
Only one TCP connection will be used for the pipelined connection (browser can send requests for all images one after another in a single RTT).

Server returns these images at a rate of 20,000 Bytes per RTT. Total size of images is 80,000 Bytes. Server can send 20,000 Bytes in one RTT therefore it requires **4 RTTs** to send the images.

[1M for to mention only the correct number of RTTs required and 1M for giving proper reasoning in support of your answer]

Q.2 a) Suppose application A is using a stream socket to transfer data to application B on a remote host. Suppose application A writes data into the socket buffer ten times. [3+4=7M]

i) Why might the underlying TCP implementation transmit *more* than ten data packets (not including retransmissions of lost packets, or control packets like SYN, ACK, or FIN)?

Sol: If the application write more than an MSS of data, a single write call may lead to multiple TCP segments. [1/0]

ii) Why might the TCP implementation transmit all of the data in fewer than ten data packets?

Sol: If the application write less than an MSS of data, the data from multiple write calls may be combined into a single TCP segment. [1/0]

iii) Why can application A safely perform a close() on the socket when it is done writing the data, even if application B has not yet received all of the data?

Sol: The TCP implementation buffers the data and continues to transmit/retransmit the data to the receiver as required. [1/0]

Q.2 b) Host A and Host B made a TCP connection and disconnected it immediately without doing any data transfer. In this process the TCP connection state of host B changed in the following sequence (read from left to right).

CLOSED	LISTEN	SYN_RCVD	ESTABLISHED	FIN_WAIT_1	TIME_WAIT	CLOSED

i) Write the sequence of states to which the Host A went through.

Sol: Sequence of state for Host A are: [1M to identify first three states correctly and 1 M for remaining two states]

CLOSED	SYN_SENT	ESTABLISHED	LAST_ACK	CLOSED

ii) For each state, write the names of TCP messages that are sent or received (or both) by Host B which triggered a state transition.

Sol: [1M to correctly identify messages up-to first four state and 1M for remaining two states messages]

CLOSED to LISTEN: Socket creation by the application
LISTEN to SYN_RCVD: Receive SYN packet and send SYN_ACK
SYN_RCVD to ESTABLISHED: Receive ACK
ESTABLISHED to FIN_WAIT_1: Send FIN
FIN_WAIT_1 to TIME_WAIT: Receive ACK and FIN together, Send ACK
TIME_WAIT to CLOSED: Timer expires

Q.3 Suppose we have two hosts, A and B. The host A wants to fetch data messages from B according to the following conventions.

- When the transport layer at A receives a request from the application layer to get the next data (DATA) message from B, A must send a request (REQ) message to B.
- Only when B receives a REQ message, it can send a DATA message back to A.
- A should deliver exactly one copy of each DATA message to the application layer.
- REQ messages can be lost (but not corrupted); DATA messages, once sent, are always delivered correctly.
- The delay along both channels (i.e., A to B and B to A) is unknown and variable.
- Assume no reordering of packets occurs in the channel.

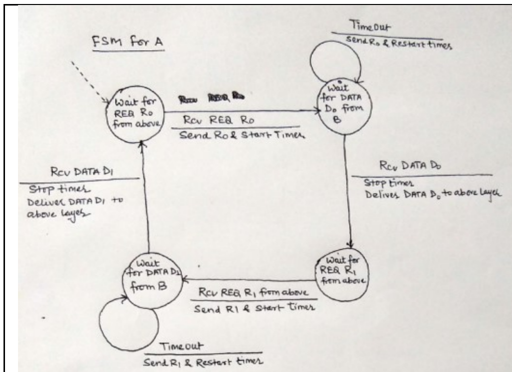
Design an FSM description of the protocol as described above. Use only those mechanisms that are absolutely necessary. [7M]

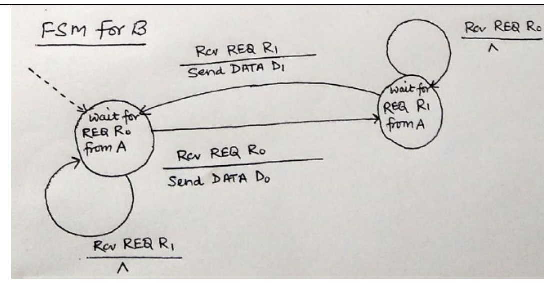
Sol: Timer, Sequence numbers are required.

[3M for handling in order data delivery and satisfying protocol semantics as REQ and RESPONSE (Which means data should not be delivered by B without a request received from A)]

[4M for working of protocol under no REQ loss condition]

[Penalty for 0.5 to 1 M for each mistake (i.e. adding unwanted things, missing timer initialization and stop, duplicate data delivery from B, initial state not mentioned]





Q.4 a) Consider the network topology shown in Fig.1. Assign network addresses to each of these 6 subnets, with the following constraints: (Specify subnet addresses in the form a.b.c.d/x) [6+2=8M]
All addresses must be allocated from the block 214.95.110.0/25

- Subnet A should have enough addresses to support 62 hosts
- Subnet B should have enough addresses to support 30 hosts
- Subnet C should have enough addresses to support 22 hosts
- Subnets D, E and F each require 2 addresses

Q.4 b) Now using the address assignments from 4(a), provide the forwarding table (in a.b.c.d/x) for the R1 router.
Sol: One of the possible solution is as follows: [1M for each subnet]
Subnet A: 214.95.110.0/26 (Ranges from 214.95.110.0 to 214.95.110.63)
Subnet B: 214.95.110.64/27 (Ranges from 214.95.110.64 to 214.95.110.95)
Subnet C: 214.95.110.96/28 and 214.95.110.112/29 (Ranges from 214.95.110.96 to 214.95.110.119)
Subnet D: 214.95.110.120/31; Subnet E: 214.95.110.122/31; Subnet F: 214.95.110.124/31

Forwarding table at Router R1 [0.5 for each correct entry]

Prefix
214.95.110.0/26
214.95.110.64/27
214.95.110.96/28
214.95.110.112/29

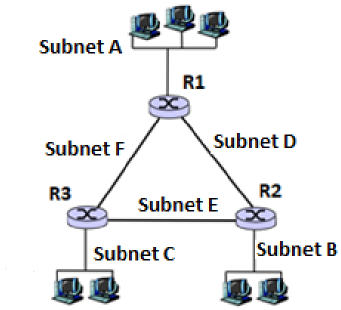


Fig.1

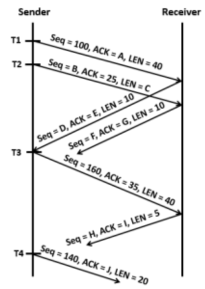


Fig.2

Q.5 a) Calculate the missing sequence numbers (Seq), acknowledgment numbers (ACK), and segment length (LEN) (i.e., **A** to **J**) in the TCP connection as shown in the Fig.2. Consider following assumptions: [5+1=6M]

- No timeouts occur at the receiver.
- The sender starts the timer at T1.
- There are no delayed acknowledgements at the sender or the receiver.

Sol: A = 25, B= 140, C=20, D=25, E=140, F=35, G=160, H=45, I=200, J=35 [0.5 for each value]

Q.5 b) Calculate the value of timeout for the segment with the sequence number 140.

Sol: The packet with sequence number 140 was originally sent at time T1 and retransmitted at T4. Therefore timeout value is T4-T2. [0/1M]

Q.6 Suppose you have registered the domain name "compnet.bk.edu" with the registrar. Assume bk.edu is an authoritative DNS server for the registered domain name. [2+2=4M]

a) Which type of DNS resource record(s) are required to be stored in the DNS hierarchy so that people can visit your web site? Show the four tuple DNS record format for each type identified by you.

Sol: Type **NS** record for your domain at the TLD server edu [compnet.bk.edu, bk.edu, NS, TTL]
Type **A** record for ***bk.edu*** server at the TLD server edu [bk.edu, IP address of bk.edu, A, TTL]
Type **A** record for ***www.comnet.bk.edu*** at the ***bk.edu*** server [www.compnet.bk.edu, IP address of your host name, A, TTL] [1M for TLD server records and 1M for authoritative server record]

b) Suppose a client access your web server www.compnet.bk.edu first time. List the steps to resolve the IP address of your domain starting from a client sending DNS query to the local DNS server. Assume DNS queries are recursive at root server. You can assume DNS entries are not cached anywhere.

Sol: Client browser sends DNS request to resolve IP address to local DNS server. [1/2M]
Local DNS server forwards the request to one of the root DNS server.
The root DNS server sends the request to one of the TLD servers responsible for edu.
The edu TLD server (i.e., the server responsible for edu domain) forwards the query to bk.edu DNS server.
The bk.edu server responds to edu TLD DNS server with the IP address of compnet.bk.edu.
The edu TLD server responds root DNS server.
The root server responds to the local DNS server with the IP address of compnet.bk.edu.
Finally, local DNS server responds back to the client machine.

****