Problem 1: Multiple choices (2 points each). Select all the correct answers from the five choices.

1. Suppose TCP congestion window size is 12 segments, and its receiver's advertised window size is 10 segments. What is the maximum number of back-to-back packets TCP can transmit in its reliable data transfer?

   - Your answer A (A) 10; (B) 11; (C) 12; (D) 20; (E) 22.

2. Which of the following protocol uses UDP?

   - Your answer C (A) HTTP; (B) FTP; (C) DNS; (D) SMTP; (E) BitTorrent.

3. Which protocol can never be used when Bob uses his new laptop computer to send and receive his email via Outlook for the first time?

   - Your answer D (A) SMTP; (B) DNS; (C) POP3 or IMAP; (D) FTP; (E) TCP.

4. Which header field does appear in one but not both UDP and TCP packet headers?

   - Your answer D, E (A) Source port number; (B) Destination port number; (C) Checksum; (D) Sequence number; (E) Acknowledgment number.

5. Which is not a feature of packet switching?

   - Your answer C (A) statistical multiplexing; (B) no reservation is needed in advance; (C) providing delay guaranteed services; (D) more efficient for bursty data traffic; (E) congestion may occur in the network.

6. Which mechanism is not required to ensure reliable data transfer?

   - Your answer B (A) error detection via checksum; (B) automatic error correction for corrupted packets; (C) retransmission upon timeout; (D) sequence numbers for transmitted packets; (E) acknowledgment numbers for received packets.

7. What are the bad effects without TCP flow control?

   - Your answer C (A) packets are always dropped by the network; (B) packets do not need to be retransmitted; (C) buffer overflows at the receiver; (D) reduced delay for packet delivery; (E) packets will take better delivery paths to reach the destination.

8. Which layers in the protocol stack are NOT typically implemented at routers?

   - Your answer A, B (A) application layer; (B) transport layer; (C) network layer; (D) link layer; (E) physical layer.

Problem 2 (3 points each): Answer the following questions. Be brief and concise.

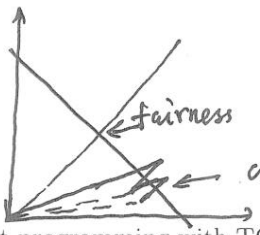
1. Web caching can reduce user-perceived response times, but cache may become stale over time. How does HTTP allow for a cache to verify that its objects are up to date?

conditional GET message: (1) request message uses the GET method; (2) the request message has an If-Modified-Since: head line

2. A UDP receiver computes the Internet checksum for the received UDP segment and finds a *mismatch* with the value carried in the checksum field. Can the receiver be absolutely certain that bit errors have occurred with the received UDP data? Briefly justify your answer.

No, the checksum field can be corrupted but the UDP payload is correct.

3. Consider two TCP connections sharing a single link, with identical round-trip-times and segment size. It is well known that the additive-increase, multiplicative-decrease (AIMD) mode can ensure fair throughput for both TCP connections eventually. Now some one claims that multiplicative-increase, additive-decrease (MIAD) can also ensure fair throughput eventually for these two connections, starting from an arbitrary window size. Show why this is NOT true. You can draw a figure to help your explanation.



diverge from the fairness point

4. Briefly explain the main steps for socket programming with TCP on the server side. You do not need to list the detailed function calls.

1. open a STREAM socket
2. bind a name to a socket
3. listen for connections on a socket
4. accept a connection on a socket
5. read and write data on a socket
6. close the socket

5. Which HTTP operation model consumes the largest amount of server resources, nonpersistent but with parallel TCP connections, persistent connections with pipelining? Briefly justify your answer.

nonpersistent with parallel TCP connections, since they need OS resources for TCP.

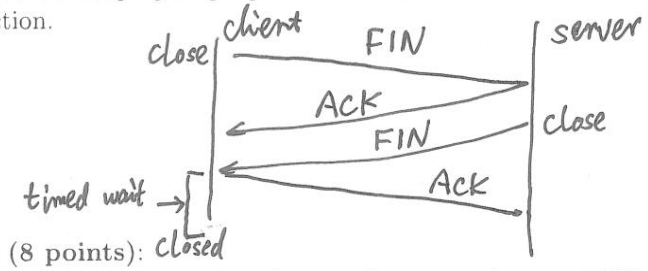
6. Specify two scenarios in reality where the local DNS server does not need to send the query to the root DNS servers.

1. local DNS server has the cached result of the IP address
2. local DNS server has the IP addresses of TLD servers

7. A group of friends decide to develop a new Internet social networking application running on their own computers, which are on and off all the time. Which model the application should use, the client-server paradigm or the peer-to-peer one? Briefly justify your answer.

peer to peer, since a server has to be on all the time while p2p does not have this requirement

8. Describe the step-by-step operation on both the client and the server when closing a TCP connection.



Problem 3 (8 points): Joe is writing programs with a client and a server that use TCP stream sockets. The following is the CLIENT code that Joe wrote. Can you help Joe to fill in the, missing parts in his code ? You can use the Appendix for references.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

int main(int argc, char *argv[]) {
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;

    char buffer[256];
    if (argc < 3) {
       fprintf(stderr,"usage %s hostname port\n", argv[0]);
       exit(0);
    }
    portno = atoi(argv[2]);

    sockfd = socket(AP_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
         (char *)&serv_addr.sin_addr.s_addr,
         server->h_length);
    serv_addr.sin_port = htons(portno);

    if (connect(sockfd, (struct sockaddr *)& serv_addr, sizeof (struct serv_addr)) < 0)
        error("ERROR connecting");

    printf("Please enter the message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);
```

```
    n = send(sockfd, buffer, 256, 0);
    if (n < 0)
         error("ERROR sending with socket");
    bzero(buffer,256);

    n = recv(sockfd, buffer, 256, 0);
    if (n < 0)
         error("ERROR receiving from socket");
    printf("%s\n",buffer);
    return 0;
}

void error(char *msg) {
    perror(msg);
    exit(0);
}
```

Problem 4 (10 points): You are asked to compute the retransmission timeout (RTO) for TCP. The initial estimated round-trip time (RTT) is set as 400ms, and initial RTT variation is 200ms. The RTT samples for 3 TCP segments are 400ms, 600ms, 400ms. In these 3 segments, the 2nd TCP segment has been retransmitted once. Compute *all* RTO values upon receiving *each* of three TCP segments. Show all the intermediate steps in your calculation. The following formula can be useful for your calculation:

$$EstimatedRTT = \frac{7}{8} \cdot EstimatedRTT + \frac{1}{8} \cdot SampleRTT$$

$$DevRTT = \frac{3}{4} \cdot DevRTT + \frac{1}{4} \cdot |SampleRTT - EstimatedRTT|$$

① for 1st TCP segment

$$\text{Est RTT} = \frac{7}{8} \times 400 + \frac{1}{8} \times 400 = 400\text{ms}$$

$$\text{Dev RTT} = \frac{3}{4} \times 200 + \frac{1}{4} \cdot |400 - 400| = 150\text{ms}$$

$$\text{RTO} = \text{Est RTT} + 4 * \text{DevRTT} = 400 + 4 \times 150\text{ms} = 1000\text{ms}$$

② for 2nd TCP segment.

ignore it since it has been rexmitted

RTO = 1000ms (same)

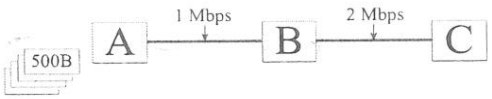
③ for 3rd TCP segment

$$\text{Est RTT} = \frac{7}{8} \times 400 + \frac{1}{8} \times 400 = 400\text{ms}$$

Dev RTT = 150ms

RTO = 1000 ms

Problem 5 (16 points): Consider sending 4 packets from Node A to Node C via Node B (see the figure below). The packet length is 500 bytes each. The propagation delay of both Link A-B and link B-C is 1 msec (0.001 second). Link A-B's bandwidth is 1Mbps ($2x10^6$ bits per second), and link B-C's bandwidth is 2Mbps.



Assume A starts transmitting the first packet at time t = 0,

1. What is the time gap between the first and second packets when they arrive at C? (i.e. the time gap between receiving the last bit of the first packet and the last bit of 2nd packet)

1. over the 1st link A-B, the time gap between the 1st & 2nd packets is $\frac{500B \leftarrow (2nd\ pkt\ size)}{1Mbps} = \frac{500 \times 8\ bits}{1Mbps} = 4ms$

2. over the 2nd link B-C, the time gap remains the same

4ms

2. When will C receive all the 4 packets?

The 1st pkt arrives at C at t =:

$$\frac{500B}{1Mbps} + 1ms + \frac{500B}{2Mbps} + 1ms = 8ms$$

↑ transmission time over A-B; ↑ prop delay; ↑ trans time over B-C

The time gap between 1st and 4th pkts:

$$\frac{500B \times 3}{1Mbps} = \frac{500 \times 8 \times 3}{1000} ms = 12ms$$

Therefore: at $t = 8ms + 12ms = 20ms$

C receives all 4 packets

Problem 6 (26 points):

1. In the following scenario, discuss how each reliable transfer protocol reacts:

   - (3 points) In the Stop-and-Wait protocol, how does the sender react when a duplicate ACK is received?

     reset the rexmit timer
     drop the duplicate ACK

   - (3 points) In the Go-back-N Protocol, how does the sender react when timeout occurs?

     send all pkts in the current window back-to-back

   - (3 points) In the Select-Repeat protocol, how does the sender react when a packet outside the window [send base, send base+window size] is received? (d)

     if the ACK is < send-base, ignore it.
     if ACK > send base + window, leap forward

   - (2 points) Identify two mechanisms to detect packet loss in reliable data transfer protocols.

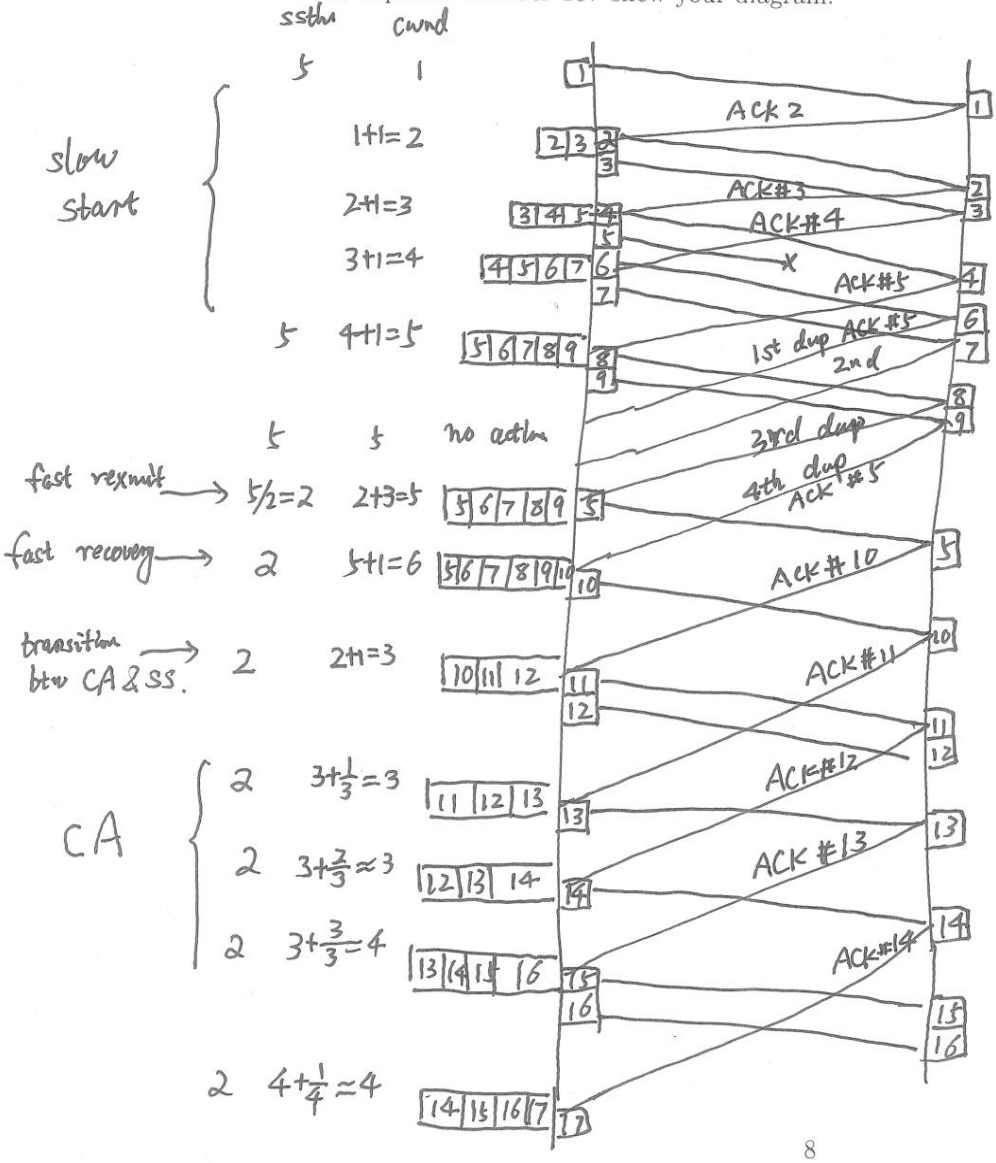
itemize

① duplicate ACKs
② timeout

2. (15 points) Consider the evolution of a TCP connection with the following characteristics. Assume that all the following algorithms are implemented in TCP congestion control: slow start, congestions avoidance, fast retransmit and fast recovery, and retransmission upon timeout. If *ssthresh* equals to *cwnd*, use the slow start algorithm in your calculation.

- The receiver acknowledges every segment, and the sender always has data available for transmission.
- Initially *ssthresh* at the sender is set to 5, and *cwnd* as 1. Assume *cwnd* and *ssthresh* are measured in segments, and the transmission time for each segment is negligible. Retransmission timeout (RTO) is initially set to 500ms at the sender and is unchanged during the connection lifetime. The RTT is 100ms for all transmissions.
- The connection starts to transmit data at time t = 0, and the initial sequence number starts from 1. Segment with sequence number 5 is lost once. No other segments are lost.

What is the congestion window size cwnd when the sender starts to transmit the segment with the sequence number 16? show your diagram.



Therefore cwnd = 4 when TCP transmits #16 pkt