Instructions:

- There are 4 questions on this exam.
- **Please use two exam blue books** – answer questions 1, 2 in one book, and the remaining two questions in the second blue book.
- Put your name and student number on the exam books NOW!
- The exam is closed book.
- **You have 80 minutes** to complete the exam. **Be a smart exam taker** - if you get stuck on one problem go on to another problem. Also, don't waste your time giving irrelevant (or not requested) details.
- The total number of points for each question is given in parenthesis. There are 100 points total. An approximate amount of time that would be reasonable to spend on each question is also given; if you follow the suggested time guidelines, you should finish with 5 minutes to spare. The exam is 80 minutes long.
- Show all your work. Partial credit is possible for an answer, but only if you show the intermediate steps in obtaining the answer.
- Good luck.

PLEASE WRITE NEATLY
I need to be able to read your answers!
Question 1: "Quickies" (28 points, 20 minutes)

Answer each of the following questions briefly, i.e., in at most a few sentences.

a) What are two important differences between a client-server and peer-peer model of communication? Answer: A client-server model requires that the server that is providing a service is always on; a peer providing a service may come and go (this is called "churn"). In Peer-to-peer, more workload (peers) also means more service capacity (peers to provide service). In client-server, more clients means more work for the server to do, so more service capacity must be provisioned.

b) If all browsers in an enterprise have a browser cache, give two reasons why it is still advantageous to have a web cache in the enterprise. Answer: An institutional cache may contain content not in the users browser – content that was requested previously by some other enterprise user. The institutional cache provides additional storage space for holding cached content.

c) What is a difference between an authoritative DNS name server and a TLD name server? Answer: The authoritative name server for a given name is the one server in the Internet that must know the name-to-IP-address translation for that name. A TLD name server will know the IP address of name servers for names in that domain. E.g., a .edu server will know the IP address of name servers for the various .edu names.

d) Suppose that a TCP client using local port number 5555 and IP address 128.119.40.186 connects to a web server (port 80) at IP address 75.74.73.72. As a result, a new server-side socket is created for communication between client and server. When the client sends the HTTP GET message destined to this new socket, what are the source and destination IP addresses and port numbers on the IP datagram carrying this HTTP GET message? What are the source and destination IP addresses and port numbers on the IP datagram carrying the server-to-client reply to this HTTP GET message? Answer: For client-to-server message the source IP, source port, dest IP, dest port: 128.119.40.186, 5555, 75.74.73.72, 80. For server-to-client reply message the source IP, source port, dest IP, dest port: 75.74.73.72, 80, 128.119.40.186, 5555.

e) What is meant by the term “encapsulation”? Answer: taking information from the layer above, and adding a header to make a new protocol data unit for this later of the protocol stack.

f) What is the purpose of the IF-MODIFIED-SINCE field of an HTTP GET message? Answer: for the client to let the server know that it has a cached copy of a piece of content that was received at the specified time following the IF-MODIFIED-SINCE string.

g) Suppose that Alice wants to send an email message to Bob. This will involve four entities: Alice’s mail client (for email composition and sending), Alice’s outgoing mail server, Bob’s incoming mail server, and Bob’s mail client (for email retrieval and viewing). Between which of these four entities does the SMTP protocol operate? What about the IMAP protocol? Answer: Alice-to-Alice’s server: SMTP, Alice’s mail server to Bob’s mail server: SMTP, Bob’s client retrieves mail from Bob’s mail server using IMAP.
Question 2: Web Content Access (20 points, 20 minutes)

Consider the scenario shown in the figure to the right. N=10 clients and a CDN node (explained below) are in a local area network with a Gbps (1,000,000,000) path with a 1 msec RTT between any two hosts in the LAN. The client LAN is connected to another network containing an origin server, cnn.com, via a 1 Mbps (1,000,000 bps) link between routers R1 and R2 with a 100 ms RTT delay. All web pages being served by cnn.com are 500Kbits in size. You can assume that any other messages involved below are very short, and thus have a transmission time of zero, but do experience propagation delays.

a) Suppose that the 10 clients all make requests to the cnn.com server. If all clients make requests at the same rate, and ignoring web browsers and web caches, what is maximum rate that CNN content (in pages per second) can be delivered to a single client. Answer: 100Kbps per client.

b) Now suppose the propagation delay of the R1-to-R2 link increases by a factor of 10. How does your answer to a) change? Answer: no change.

c) Suppose a client makes an HTTP request to the cnn.com web server. How much time is required from when the user first enters the cnn.com URL into a browser and when the page is received at that client. Ignore browser caching. You need not consider DNS delays but should take into account delays involving transport layer messages that are sent. Answer: 202 msec propagation delay for TCP handshake with cnn.com server. 202 msec propagation delay for HTTP GET and reply cnn.com server. 500 msec transmission delay on the 1Mbps link + .5msec transmission delay on each of the LAN links. Total delay: 904. msecs

The CDN (content distribution network) node in the client LAN contains copies of a fixed set of pages from the CNN web site (intentionally placed there by the CDN). If a client requests content from the CNN server (via a traditionally HTTP GET) that is available in the CDN node, the CNN server will respond to the GET request with a short “redirection” reply that tells the requesting client to get the content from the CDN node.

d) Suppose that the CDN node, which can hold x pages of content, is filled with the most popular content on the cnn.com site, and the result is that 50% of the requests made to the cnn.com server are for content that is available in the CDN node. How does you answer to a) change? Answer: Since only half the pages need be delivered over the bottleneck link, the rate at which pages from cnn.com (and its CDN node) is twice as large as in a)

e) Now suppose that the CDN node is replaced by a traditional web cache that can also hold x pages (where x here is the same x as in d)). Assuming the client request characteristics are unchanged from d), would you expect that the fraction of time that content is found in the web cache is greater than, equal to, or less than 50%? Briefly explain your answer. Answer: the fraction of time that the content would be found in the cache would be less because sometimes unpopular content would be requested,
causing content that is more popular to be flushed out of the cache, causing more misses.

**Question 3: A protocol for merging two received values (30 points, 25 minutes)**

Consider the diagram to the right, which shows three nodes, A, B, and C. A and C are connected by a bi-directional channel with variable and known delay; B and C are connected by a separate independent bi-directional channel. A and B can not communicate with each other. The A-to-C and B-to-C channels may corrupt but will not lose messages. The C-to-A and C-to-B channels are perfect (they will not lose or corrupt messages). Advice: re-read the channel models and make sure you understand them!

You are to design a protocol that operates as follows. A and B each receive data from their upper layer, via a call to `rdt_send(data)`, which returns the data to be sent, exactly the same as we studied in the `rdt` protocols in class and in the text. A and B are to reliably send the sequence of data obtained via subsequent calls from `rdt_send(data)` to C. Whenever node C has received a data value each from A and B, it will then pass the two values up to the upper layer at C at the same time via a call to `deliver_data(data_from_A, data_from_B)`. Node C’s job should thus pair together the `ith` data item from A with the `ith` data item from B, and to deliver this pair (exactly once, with pairs being delivered in their proper order) to the upper layer at C. Node A and node B should NOT begin sending the next data item until they know that C has delivered the current data item (paired with the current data item from the other node).

a) Give a short description in words of how A and C operate (I’ll assume that your protocol for B is essentially the same as A, so you do not need to discuss B itself. But you’ll want to discuss B’s behavior in describing C of course). Make sure to describe the purpose of each type of message used by your protocol (from A to C, and from C to A) and any fields within a message. **Answer:** in my protocol, A will send the next message to C after receiving an ACK. If a NAK is received, A will resend the current packet. At C, the key insight is that C will send an ACK to A and B at the same time only after it has received a packet from A and B. C will have to remember whether it is waiting for (i) a message from both A and B, (ii) a message from A, having already received a message from B; or (iii) a message from B, having already received a message from A.

b) Briefly describe (a sentence each) why you use (or do not use) the following reliability mechanisms: (i) checksums, (ii) ACKs and/or NACKs, (iii) timers, and (iv) sequence numbers. **Answer:** a checksum is used to detect bit errors. An ACK or NACK is used by C to tell A whether or not the message received from A was OK (ACK) or had an error (NAK). No need for timers here since there is no packet loss – a node waiting for a packet will eventually get that packet. No need for sequence numbers either.

c) Specify the protocol for implementing the data transfer described above for node A and node C (again, presumably node B will operate in a manner similar to A) using a
finite state machine. Your protocol should only use the mechanisms needed to correctly implement the protocol. Answer:

Now consider the case that the A-to-C and B-to-C channels may both corrupt and lose messages.

d) Specify a protocol for implementing the data transfer described above for this corrupting and lossy channel for node A and node C using a finite state machine. If you choose to use sequence numbers you need not worry about using a small sequence number space. Answer: my solution here uses a variable currentseq# at the sender and receiver to indicate the sequence number of the data item currently being sent (by A) or being waited for (at C). A timer is used at the sender to retransmit a message that may have been lost. It is interesting to note that C does not have to respond to old messages with an ACK, since it has delivered an ACK for that packet, which will eventually get to A correctly. Consequently, A will also never receive ACKs for old messages.
Question 4: Pipelined Protocols  (22 points, 15 minutes)

In this question we’ll explore several aspects of pipelined protocols (Go-Back-N and Selective Repeat).

a) Suppose that a sender is connected to a receiver over a 1 Gbps (1,000,000,000 bits per second) link, with an RTT propagation delay of 10 milliseconds. Packets are 10,000 bits long. What size window is needed to ensure that this link’s utilization is at least
50 percent? Answer: A packet takes $10^{-5}$ secs to transmit, so at full utilization, a sender can send 1000 packets in 10 msec. At 50% utilization, that would thus be 500 packets.

b) Consider a window of size $N$ and the Go-Back-N protocol. Suppose the current sender window at time $t$ starts at sequence number $x$ (i.e., that the sender window covers sequence numbers $x$ to $x+N-1$). Is it possible that the receiver window at time $t$ start at (i) a sequence number less than $x$, (ii) equal to $x$, or (iii) greater than $x$? Explain your answers briefly [Hint: more than one of cases (i) – (iii) may be true]. Answer: (i) can’t happen. (ii) suppose every packet sent by the sender has been ACKed by the receiver and all ACKs have been received by the sender. In this case, the windows are aligned. (iii) Let $x$ be the smallest sequence number of a packet sent by the sender and received and ACK by the receiver, but for which the sender has yet to receive this ACK. In this case, the sender window beings at $x$, and the receivers window is larger than $x$ (since $x$ has been ACKed).

c) Does your answer to b) change for the case of Selective Repeat? Answer: No. The arguments above hold in both cases.

d) Consider the GBN protocol and assume that the window size, $N$ (which you can choose), is equal to the size of the sequence number space. Given an example (in the form of a sender/receiver timing diagram showing data and ACK messages being exchanged) showing that the receiver may deliver duplicate copies of data to the application layer above. Answer: see below for the case of $N=2$, for d) and e):

e) Again consider the GBN protocol and assume that the window size, $N$, is equal to the size of the sequence number space. Given an example (in the form of a sender/receiver timing diagram showing data and ACK messages being exchanged) showing that the receiver may not deliver some of the data sent by the sender to the application layer above. Answer: can use answer from 2d above.