Instructions:

- There are 3 questions on this exam.
- 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" (36 points (4 each), 30 minutes)

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

a) We’ve said that the Internet is a network of networks. What does that mean? Answer: The Internet is made up of a large number of tier-1, tier-2 and tier-3 networks. Each network is owned and operated and controlled by a different network owner/operator.

b) 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.

c) Consider a network in which all users need to send data at a constant rate for a long period of time, and if they aren’t able to achieve this constant rate throughout, application functions poorly. Is circuit switching or packet switching better suited for this scenario? Explain. Answer: circuit switching is better here, since there will be no resource contention, and resources allocation (capacity) will always be in use.

d) 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.

e) What is the purpose of a manifest file in dynamic adaptive streaming (e.g., as done in Netflix)? Answer: The manifest file contains information about various time segments (chunks) in the video – the URL where a given chunk of video, encoded at a given rate can be found. A Netflix client obtains a manifest file before starting to receive the streamed video (indeed it receives the streamed video by requesting chunks in the manifest file).

f) 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 than 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.

g) Consider a popular domain name like madonna.com. Does the first person who registers the domain name have the right to use the name (e.g., run a web server that serves pages from Madonna.com) under all circumstances? Briefly explain your answer. Answer: in order to be able to be assigned a domain name, the owner must provide an Internet service that is somehow related to that name. So a company called Madonna Dresses that sold generic women’s clothing have gotten madonna.com if it were to serve web pages for its company. However, someone with no association with anything madonna-esque could not get that domain name, while not using it for any madonna-esque
h) Give two important differences between the UDP and TCP transport-layer protocols. \textit{Answer: UDP provides no error control, congestion control, or flow control, while TCP provides all three.}

i) What is meant by a third party cookie? \textit{Answer: A third party cookie is a cookie returned to a web client when the web client fetches a piece of content (e.g., an image or an ad) from server Y that was referenced in a base page on server X that was originally fetched by the client. Server Y is the “third party.”}

\textbf{Question 2: A protocol for delivering two received values (33 points, 25 minutes)}

Consider the figure to the right, which shows three nodes, A, B, and C. A and C are connected by a bi-directional channel; B and C are connected by a separate independent bi-directional channel. A and B can not communicate with each other.

The protocol between A and C (and B and C) operates as follows. A and B each receive data from their upper layer, via a call to \texttt{rdt\_send(data)}, which returns the data to be sent, exactly the same as the \texttt{rdt} protocols we studied in class and in the text. A and B are to reliably send their sequence of data obtained via subsequent calls from \texttt{rdt\_send(data)} to C. Whenever node C has received a data value \textit{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 \texttt{deliver\_data(data\_from\_A, data\_from\_B)}. Node C’s job is thus pair together the \textit{ith} data item from A with the \textit{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).

I’ve done some of the work for you! The figures below show the finite state machine representation of the protocols at A and C (B is essentially the same as A) that will operate over \textit{PERFECT} channels (from A-to-C, C-back-to-A, B-to-C and C-back-to-B) that do not lose or corrupt messages. In my protocol, A sends a message to C and then waits for an ACK from C before sending the next message to C. Note that C sends an ACK to A and B when it has received a message from both A and B and thus delivered both to the upper-layer application at C; the ACK here thus serves as a “Go-Ahead” message for A so it can get and send the next data item; the ACK here doesn’t really serve as an “I got your message correctly message”, since all channels are perfect, so A knows that any message it sent was received correctly at C.
Now consider the case that the A-to-C and B-to-C channels may corrupt but will not lose messages. If C receives a corrupt message, it knows whether that message was sent by A or B. The C-to-A and C-to-B channels are still perfect (they will not lose or corrupt messages). You should modify my protocol above to provide reliable communication and delivery of paired-data at C for this case that the A-to-C and B-to-C channels may corrupt but will not lose messages:

a) In one sentence each, describe the structure (fields) and purpose of each type of message used by your protocol (from A to C, and from C to A). You don’t need to say anything about B here, since it essentially identical to A. Answer: the data message from A to C contains the data received from above and a checksum. The ACK or NACK messages are used by C to let A know if the message last sent was received correctly or not.

b) Briefly describe (a sentence or two 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 in your protocol. Answer: a checksum is needed to detect whether a packet is corrupted or not. ACKs and NAKs are used by C to tell A whether or not a data packet from A was received correctly. Timers are not needed – there is no loss so every message sent will eventually pop out the other end of the channel, and which point another message will eventually be sent in the opposite direction. Sequence numbers are not needed because any retransmission by A will be in response to a NAK from C, so C will know that the next message received after it sends a NAK will be a retransmission of the message it NAKed.

c) Describe in a few sentences how A and C operate. Your protocol should only use the mechanisms needed to correctly implement the protocol. A simpler, more minimalist protocol is preferred over a more complex protocol. Answer: A sends the next message to C initially (once) and then after every ACK. If a NAK is received it retransmits the last message it send. C waits until it has correctly receives a message from both A and B, and then ACKs them both. Note that if C were to ACK after receiving just one message say from A, it would have to buffer the next message from
A if no message had yet been received from B. C sends a NAK to the sender if the message received is corrupted.

d) Specify your protocol by modifying/extending my finite state machines for A and C above.
Question 3: A web server and ad server scenario (31 points, 25 minutes)

Consider the scenario shown in the figure below in which a client wants to access a web server. The web server is connected to the Internet by a link with a transmission capacity of 1,000,000,000 bits (a gigabit) per second.

a) Suppose a web page (including all of its images, which are also stored on the web server) is 1,000,000 bits long. How long does it take for the server to send a web page (including all of its images) into the Internet over the gigabit (1,000,000,000 bits per second) link? Answer: time to send = 1,000,000/1,000,000,000 = .001 secs.

b) What is the maximum number of web pages that the web server can transmit per second, assuming all of the web pages (including all of its images, which are stored on the web server) are the same size as in (a)? Answer: 1000 pages per second.

Now let’s consider the case that the web server serves a base page, but that the base page has three advertisements, each of which is served by the ad server shown in the figure above.

c) Suppose now that (i) the base web page takes X1 seconds to transmit into the Internet, and that the client-to-web-server RTT is RTT1, (ii) each advertisement takes X2 seconds to transmit into the Internet, and that the client-to-ad-server RTT is RTT2. How much time is taken from when the client first clicks on the link to access a web page containing these three ads until the page is displayed? Your answer should be in the form of a formula involving X1, RTT1, X2, and RTT2. Briefly explain how you arrived at your formula. You should remember that HTTP runs over TCP. You do NOT have to worry about DNS delays here. You can assume that “small” messages (i.e., messages that don’t contain a web page or an image) take zero time to transmit into a link, but do experience a propagation delay. Your answer should be in the form of a formula involving X1, RTT1, X2, and RTT2. Briefly explain how you arrived at your formula. Answer: note that we will need to account for the TCP handshake delays. To fetch the base page requires 2RTT1+X1 (one RTT1 for the TCP handshake and one RTT1 for the HTTP GET request and reply, and X1 time for the server to actually transmit the file into the link). Each ad similarly takes 2RTT2+X2, assuming non-persistent HTTP. So the overall delay is 2RTT1 + X1 + 6RTT2 + 3X2.

d) Now suppose that the client’s browser has a cache and the client has previously visited the web page. The web page at the server has changed since the client last viewed the web page, but the advertisements have not changed. Under the otherwise same assumptions as (c), how much time is taken from when the client first clicks on the link to access a web page containing these three ads until the page is displayed? Your answer should again be in the form of a formula. Briefly explain why your formula here differs from your answer to (c). Answer: one RTT2 is still needed to verify that each ad has not been modified (i.e., the client needs to send an IF-MODIFIED-SINCE HTTP GET to the ad server, and the ad
server needs to respond with a 304 Not Modified. But the ad files themselves do not need to be transmitted. So the overall delay is $2RTT1 + X1 + 6RTT2$.

e) Let’s reconsider (c) but now also account for DNS delays. Assume that the local DNS cache is empty. Assume that the time RTT between the client and the local DNS server is $D1$, and that the time needed (including all message transmission and RTT times needed for the local DNS server to resolve a request through the root, TLD and authoritative name servers is $D2$. How much time is taken from when the client first clicks on the link to access a web page containing these three ads until the page is displayed, including the DNS delays? Briefly explain how you arrived at your formula. Answer: A DNS lookup is need to find the IP address of the web server. This will take time $D1+D2$, since the local cache is initially empty. The first ad request will also take $D1+D2$. The next two ad requests, however will only take $D1$, since the local DNS server will have cached the name-to-IP-address translation for the ad server. Thus the total delay is the delay in part (c) plus $4D1 + 2D2$. 

Please turn over – there’s text on the other side!