Problem 1: (16 points; 10 minutes) Give very short answers to the following:

1. True or False: Framing is the problem of breaking data into frames so that it can be transported efficiently.
2. True or False: the minimum spanning tree algorithm used in learning bridges is used to remove the count to infinity problem.
3. The key ideas in layering are ——— and ———.
4. True or False: ATM runs over IP.
5. Fill in the blank: ——— sets up a virtual circuit before starting to forward packets.
7. True or False: Fragmentation is not carried out by layer 2 switches.
8. True or False: Dynamic link costs in routing can cause oscillation.

Problem 2: (20 points; 15 minutes) For four of the following alternatives, describe one situation where the first alternative is better, and another where the second is better.

1. ARQ and Forward Error Correction
2. Source Routing and Datagram
3. CRC and Internet Checksum
4. 4b/5B and Manchester Encoding
5. Distance Vector and Link State Routing

Problem 3: (12 points; 10 minutes)

1. (4 points) Why are network caches (e.g., ARP cache, or learning bridges table for remembering seen neighbors) used for?
2. (8 points) What are the disadvantages of using network caches? Show this using two different examples of how the ARP cache can cause problems; in each case describe what the observed effect is.

Problem 4: (36 points; 25 minutes) Use an example to illustrate the following.

1. (4 points) Bad effect of timeout being picked too large in Go-Back-N. Use a window size of 4.
2. (4 points) Bad effect of timeout being picked too small in selective repeat. Use a window size of 2.
3. (4 points) Learning bridges being more efficient than repeaters. Use a loop free topology (i.e., network).
4. (6 points) Routers (layer 3) being more efficient than learning bridges (layer 2). Use a topology with a loop.
5. (6 points) The count-to-infinity problem. The loop should be made of 4 nodes.
6. (6 points) A solution to the count-to-infinity problem using the same topology as (5).
7. (6 points) A good case and a bad case for a reservation based MAC protocol.

Problem 5: (16 points; 15 minutes) Consider the following problems:

a. (8 points) We want to track not only the shortest path between nodes, but all paths that satisfy a certain criteria (say length less than x hops). Which is more appropriate: distance vector or link state? Discuss.

b. (8 points) Consider ARQ design for a connection with a variable/high delay. The timeout value is typically picked to be large – why? As an optimization, you decide to improve the scheme by adding negative acknowledgements – if you detect at the receiver that a packet is corrupt or missing you send a negative acknowledgement. Note that we keep the normal timeout/ACK scheme in operation as well. Discuss this idea.