Section A: True of False (2 points each)
State True or False and give a very short (sentence fragment) explanation for why.

1. The IP address is a de-multiplexing key. True or False

2. The capacity of a link depends on its bandwidth and signal-to-noise ratio. True or False

3. A code with a Hamming distance of 5 detects all 1, 2, 3, and 4 bit errors. True or False.

4. CSMA works well for networks with a large bandwidth-delay product. True or False

5. ISP routing policy decisions help to make the Internet scale. True or False

6. The end-to-end argument is a performance optimization methodology. True or False

7. One-way latency depends on the bit rate of the link. True or False

8. The Ethernet address of a device changes over time. True or False

Section B: Short answers (2 points each)
Give a very short (single sentence) answer to the question.

1. What is the main different between CRCs and ARQ?

2. Name three fields in the 802.11 header.

3. Why is there a need for the new IPv6 standard when the IPv4 is available?

4. Why is the RTS/CTS protocol often not used in practice?

5. In what way is the link state protocol an improvement over distance vector?

6. Why is a spanning tree algorithm important for LAN switches?
Section C: Longer Answers (10 points each)
1. Consider the arrangement of hosts and learning bridges in the figure below. Assume the spanning tree algorithm has run and all bridges forwarding tables are initially empty.

a) Draw the final spanning tree, on top of the diagram or otherwise.

b) Give the contents of the forwarding tables learned by bridges B1, B2 and B3 after the following three transmissions: A sends to C; then C sends to A; then D sends to C. Each table should have three rows, one for A, C and D. To identify the port to reach A, C or D, use direction descriptors such as “left”, “right”, “up-right” and “down”, plus “not learned”.

Table for B1:  Table for B2:  Table for B3:

c) Suppose the spanning tree algorithm does not run at all. What will happen on the first packet transmission (A sends to C)?
[DIFFICULT] 2. You are designing a cable modem network. The network is a tree. When the head-end sends, the message goes down all branches of the tree and is delivered to all clients. When a client sends, the message goes up to the root of the tree only; it does not go down any branches and so one client cannot hear any other client. Clients talk to the head-end and vice-versa. Clients do not talk to other clients. The load on the cable network is expected to be fairly heavy, but only a small fraction of the many clients, e.g., 10 out of 100, are expected to have traffic to send at a given time.

Your job is to design a MAC protocol that makes efficient use of bandwidth by using the techniques we have seen. Consider only how to share the channel, not reliability. You may use time, frequency and code division multiplexing to statically divide the channel as part of your answer, but you must also share the channel in a way that depends on the traffic. Be sure to describe (1) how the head-end sends packets to the clients and why this is efficient; and (2) how each client sends its packets to the head-end and why this is efficient.
3. Consider a classic Ethernet and suppose there are two hosts, A and B, that have a long stream of packets to send. A and B collide, A wins the backoff (by choosing 0 when B chooses 1) and successfully sends a packet. Next A tries to send a second packet and promptly collides with B retransmitting its first packet. So A and B both backoff.

a) What is the probability that A sends before B with no further collision? We call this winning the backoff.

b) After A wins the backoff it soon collides with B again because A starts to send a third packet while B retransmits its first packet. What is the probability that B sends before A with no further collision?

c) What is the likely outcome from here on out for the next 100 (say) packets that are transmitted given that A and B both keep following the Ethernet rules and have packets to send?

d) The above phenomenon is called “Ethernet capture”. It is not much of a problem in practice. Why might this be?
4. Consider running Dijkstra’s algorithm over network shown below.

![Network Diagram]

a) What is shortest path tree rooted at B? You may draw over the figure to point out the links that are part of the tree, or draw a new figure above to the right if you prefer.

b) Draw a table that gives the confirmed nodes (with costs) and the tentative nodes (with costs) at the end of each iteration of the algorithm until the shortest path calculation is complete.

c) When there are multiple equal cost paths, as in the example above, do the nodes need to break ties the same way to construct loop-free paths? Why or why not?