In this homework, for simplicity, K means 10^3, M means 10^6, and G means 10^9. But read the green box on page 44-46 of Peterson & Davie to understand the precise meaning of these notations in different contexts. Show your work when appropriate, do not just write down a single number as your answer if you want full credit.

Problem 1 (20 points): Suppose you are designing a multi-access (i.e. broadcast) wired netwo using CSMA/CD (Carrier Sense Multiple Access/Collision Detect) mechanisms similar to t Ethernet for media access control. Assume that in your design, the network's link speed is 5Mbps, frame sizes may range from 100 bytes to 5000 bytes. Also assume that data sign propagates on the network link at a speed of $2*10^8$ meters per second.

(a) What is the maximum allowable link length (in meters) between any pair of hosts connect to the same link? Show your work. (15 points)

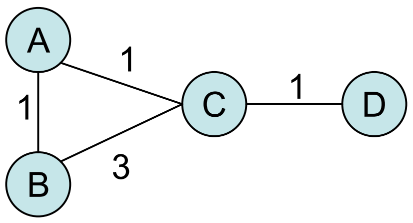
To guarantee that a collision is detected, the link length should be provisioned for the worst case scenario. The worst case is when A sends a message, immediately before A's first bit arrives at B, B starts sending and collides with A. For A to detect the collision, A must transmit until B's first bit has propagated to A, which is bounded by the transmission time of the smallest allowed packet (100B). Let d be the maximum link length, then

$2*d/(2*10^8) < 100*8/(5*10^6)$
$d < 16{,}000$ meters

(b) If the link length between two hosts is longer than this maximum, what problem may occu Only a short answer is required. (5 points)

If link length is greater than d, then a collision may not be detected, thus a collided packet may not be retried by the link layer. End-to-end correctness may still be maintained by higher level protocols like TCP, but performance will likely be reduced.

Problem 2 (20 points) One of the main problems with the distance vector routing algorithm is th
potential to form a routing loop.



(a) Based on the network shown above, describe a sequence of events that will cause a routi loop to form (and thus lead to the count-to-infinity situation). Assume no poison reverse is used, no triggered update is used (i.e. with periodic update only). (8 points)

C-D fails, C routes to A for D, A continues to route to C for D, loop created, each round DV cost increases.

(b) Poison reverse is an addition to the basic distance vector routing algorithm. The idea is, if node X uses node Y as its next hop to get to node Z, then X would advertise a cost of infir for node Z when sending a distance vector update message to node Y. Do you think poisc reserve is sufficient to prevent routing loop? Explain your answer in detail. (8 point)

No. Even with poison reverse, C can pick B as next hop for D. A loop is formed again.

(c) In the inter-domain routing protocol BGP, the route from one autonomous system (AS) to destination AS is typically chosen by picking the neighbor AS who advertises the shortest path to the destination AS as the next hop. What mechanism does BGP use to prevent routing loop? (4 points)

BGP uses path vector, explicitly states the AS path so no loop is possible.

Problem 3 (20 points) – (Peterson & Davie p.152 problem 15) Prove the Internet checksum computation shown in the text is independent of byte order (host order or network order) excep that the bytes in the final checksum should be swapped later to be in the correct order. Specifically, show that the sum of 16-bit word integers can be computed in either byte order. F example, if the ones complement sum (denoted by +') of 16-bit words is represented as

[A,B] +' [C,D] +' … +' [Y,Z]

The following swapped sum is the same as the original sum above:

[B,A] +' [D,C] +' … +' [Z,Y]

(Note: You need to show a general mathematical proof)

It suffices to show that [A,B] +' [C,D] = swap([B,A] +' [D,C]).

<u>Case I</u>: A + C and B + D have no carry (i.e. no overflow).

[A,B] +' [C,D] = swap([B,A] +' [D,C]) is trivially true.

<u>Case II</u>: A + C has carry, but B + D has no carry.

In [A,B] +' [C,D], the carry bit will be added into B + D due to 1's complement arithmetic. Any resulting carry will be added to A + C in normal addition.
In [B,A] +' [D,C], the carry bit will be added into B + D in normal addition, any resulting carry will be added to A + C due to 1's complement addition.

Thus, [A,B] +' [C,D] = swap([B,A] +' [D,C]).

<u>Case III</u>: A + C has no carry, but B + D has carry.

Same argument as Case II.

<u>Case IV</u>: A + C and B + D both have carry.

In [A,B] +' [C,D], the A + C carry bit will be added to B + D due to 1's complement addition, the B + D carry bit will be added to A + C in normal addition. No further carry is possible.

In [B,A] +' [D,C], the B + D carry bit will be added to A + C due to 1's complement addition, the A + C carry bit will be added to B + D in normal addition. No further carry is possible.

Thus, [A,B] +' [C,D] = swap([B,A] +' [D,C]).

Problem 4 (20 points) Given the following intra-domain routing table (learned via OSPF) and inter-domain routing table (learned via BGP) at a router, construct the final routing table for all destination networks learned. Your final routing table should have a column for destination network address pattern, CIDR mask, and next hop (R1…R6).

Intra-domain routing table (from OSPF)

Address Pattern	CIDR Mask	Next Hop
128.42.222.3	255.255.255.0	R1
128.42.128.4	255.255.128.0	R2
18.0.0.0	255.0.0.0	R4
128.42.127.3	255.255.248.0	R6
128.42.216.0	255.255.248.0	R5
128.42.128.4	255.255.0.0	R3

Inter-domain routing table (from BGP)

Address Pattern	CIDR Mask	BGP Next Hop	Correct next hop
128.2.111.0	255.255.255.0	128.42.60.1	R3
12.222.128.0	255.255.128.0	18.111.12.1	R4
133.0.0.0	255.0.0.0	128.42.120.32	R6
36.33.88.0	255.255.248.0	128.42.226.4	R2
73.128.222.0	255.255.248.0	128.42.124.45	R6
55.34.0.0	255.255.0.0	128.42.220.121	R5

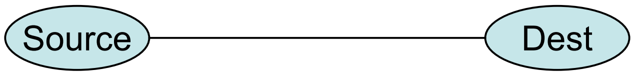
Your table should contain 12 entries with these correct next hops. The 6 entries learned from intra-domain routing remain unchanged.

We remember that, in order to find the best (more specific) match of an address (A) against the entries of a routing table with CIDR mask, you have to (& denotes the bitwise AND operation):

(1) Find all the entries for which the following is true: (A & CIDRmask_of_entry) = (Pattern_of_entry & CIDRmask_of_entry)

(2) Among the entries found at point (1), select the one for which the match is the longest (more specific)

Problem 5 (20 points): Your company is designing a specialized reliable transmission protocol that will be used to transfer data from a source node to a destination node over a private datagram network link that directly connects the source and the destination.



These are the various characteristics given to you:

1. Link speed = 1Mbps
2. Link one-way propagation delay = 245 ms
3. Data packet size = 1000 bytes (this includes packet header)
4. Bit errors are extremely rare; when it does occur, it's always only one bit error per data pack

Your job is to define part of the packet header for the data packets generated by the source. In particular, you need to define the data packet header field(s) needed to implement the most appropriate and best performing reliable transmission mechanism(s) for the above characteristics. You should choose the size of the field(s) such that the header overhead is minimized.

Note: Source and destination address fields are irrelevant. You only need to define the field(s) that are relevant to the reliable transmission mechanism(s) you choose to implement for this network.

For each data packet header field you need in your design, states: (a) the size of the field in number of bits, (b) the purpose of the field in detail, (c) why the field size you choose is the mo appropriate.

Since bit error rate is extremely low, we can immediately eliminate error correction code strategies since they will incur a high overhead. However, we still need a mechanism to detect error, and when an error is detected, cause the packet to be retransmitted.

Since there is at most 1 bit error per packet, the most efficient scheme is to use a 1 bit parity.
So, field #1 is (a) 1 bit (b) for computing the 1 bit parity of the message (c) it is sufficient to detect the error specified.

Because the network has a high propagation delay, we need a sliding window mechanism to fully utilize the network. The one way delay is 245ms, thus RTT is 490ms.
After the source has finished sending the window's first packet, it waits 490ms for the first ack. In this time it can send at most: (490ms * 1Mbps) = 61250 Bytes = at most 61 1000-byte packets. Including the first one, this is a total of 62 outstanding packets in the window. But we need at least 124 sequence numbers because the sequence numbers of 2 consecutive windows must not overlap. If they do, a packet in the second window may be mistaken for a retransmitted packet in the first window.
So, field #2 is (a) 7 bit (b) sequence number for a sliding window protocol (c) 7 bits are necessary to implement 124 sequence numbers.