EECE 7374 – Fall 2023
Homework #5 Solution

1) **CRC**
If we divide 10011 into 1010101010 0000, we get 1011011100, with a remainder of R=0100. Note that, G=10011 is CRC-4-ITU standard.
If we divide 10011 into 1001000101 0000, we get 1000100011, with a remainder of R=0101.

2) **Polling**
Node 1 must wait to be polled. In the worst case, all other nodes have Q bits to send, and the Q bits arrive to node 1 just after node 1 completes a transmission. Before node 1 gets polled again, N - 1other nodes transmit Q bits at rate R, giving a delay of (N - 1)Q/R. In addition to this, there are N polling delays. So the total wait is $(N - 1)Q/R + N\, d_{poll}$.

3) **CSMA/CD**
The node chooses K from the elements in the set {0, 1, 2, … , 15} with equal probability. The probability that it chooses K = 5 is thus 1/16. With K = 5, the node waits 5 * 512 = 2560 bit times. The corresponding delay over a 10 Mbps Ethernet link is (2560 bits)/(10^7 bits/sec) = 256 microseconds.

4) **Self-learning switching**
Node A creates a TCP SYN packet, which (after encapsulation in an IP datagram) gets encapsulated into an Ethernet frame. This Ethernet frame will have B's MAC address for its destination address. Node A transmits the frame. When the frame arrives at the switch, the switch will take note of A's location and then transmit the frame onto the other N – 1 links, giving a total of N transmissions so far. When B receives the frame, it will send a SYNACK, encapsulated in an Ethernet frame with A's MAC address for the destination address. Thus, there are N + 1 frames so far. When the switch receives the frame, it will take note of B's location; it will already have an entry in its table for A and thus will only transmit the frame onto one link. Thus, there are N + 2 frames so far. When A receives the SYNACK it will send an ACK. Two more transmissions are required for this ACK, giving a total of N + 4 transmitted frames.

5) **Multiple access protocols: voice-over-IP and data.**
   a) TDMA works well here since it provides a constant bit rate service of 1 slot per frame. CSMA will not work as well here (unless the channel utilization is low) due to collisions and variable amount of time needed to access the channel (for example, channel access delays can be unbounded) and the need for voice packets to be played out synchronously and with low delay at the receiver. Slotted Aloha has the same answer as CSMA. Token passing works well here since each station gets a turn to transmit once per token round, yielding an essentially constant bit rate service.
   b) TDMA would not work well here as if there is only one station with something to send, it can only send once per frame. Hence, the access delays are long, and the throughput over a long period of time is only 1/N of the channel capacity. CSMA would work well since at low utilization, a node will get to use the channel as soon as it need to. Slotted Aloha

has the same answer as CSMA. Token passing would work better than TDMA but slightly less well than CSMA and Slotted Aloha, since it must wait for the token to be passed to the other stations (who likely wouldn't use it) before sending again.

c) Here are two possible answers. One approach would be to divide the channel into two "pieces"-one for data packets and one for voice. This can be accomplished by assigning some number of TDMA slots for voice calls (for example, one slot to each user). Also, add some additional slots and allow the stations with data to send to perform random access (for example, slotted Aloha or CSMA) within those data slots only. A second approach would be to use token passing with priorities, and give priority to voice packets.