Henry Hoekel

14207985

Homework 2

2. Show the 4B/5B encoding, and the resulting NRZI signal, for the following bit sequence: 1110 0101 0000 0011

Message: 1110 0101 0000 0011

4B/5B: 11100 01011 11110 10101

Signal: 10111 10010 10100 11001

**5.** Assuming a framing protocol that uses bit stuffing, show the bit sequence transmitted over the link when the frame contains the following bit sequence:

110101111101011111101011111110

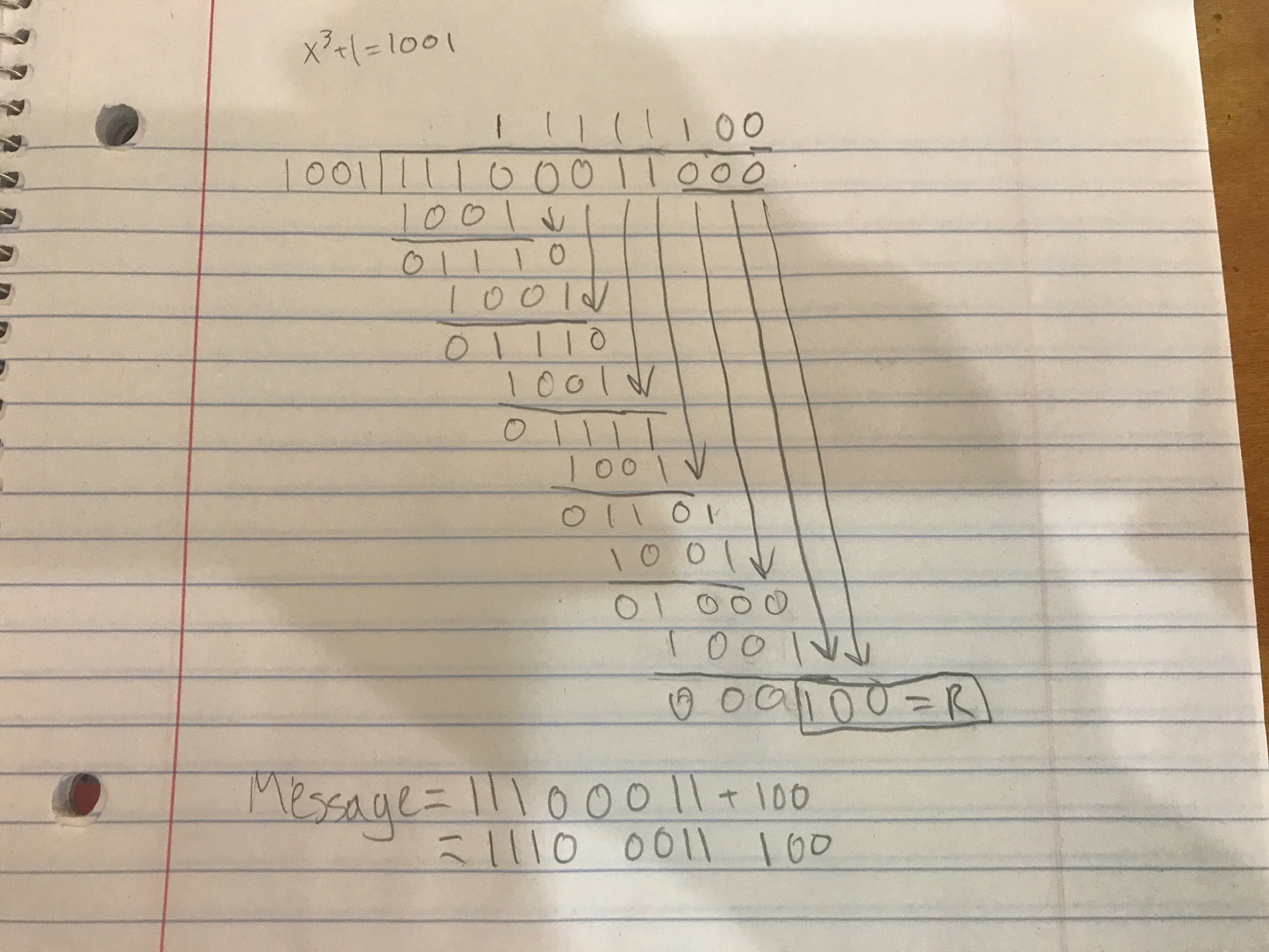
Mark the stuffed bits.

110101111100101111101010111110110

11010111110***0***1011111***0***101011111***0***110

**18.** Suppose we want to transmit the message 11100011 and protect it from errors using the CRC polynomial X^3 +1.

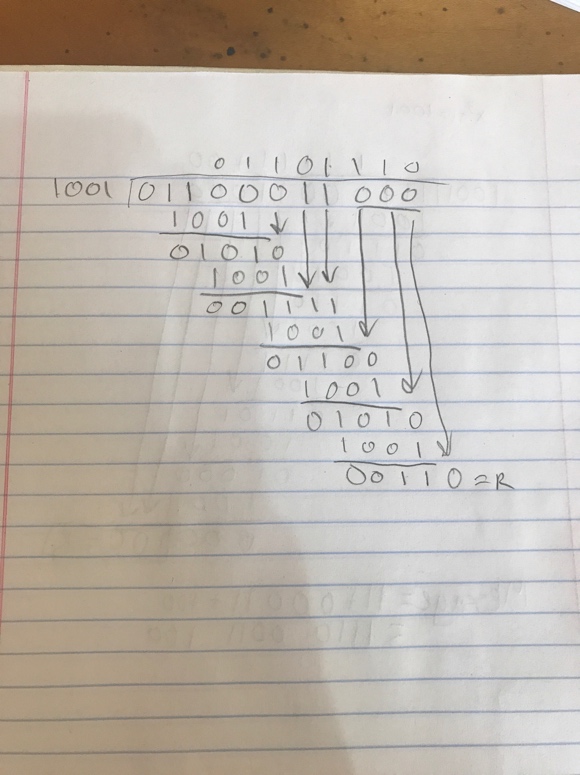
**(a)** Use polynomial long division to determine the message that should be transmitted.



After division, the remainder is 100. This appended to the original message (11100011) comes out to:

Message: 11100011100

**(b)** Suppose the leftmost bit of the message is inverted due to noise on the transmission link. What is the result of the receiver’s CRC calculation? How does the receiver know that an error has occurred?



The remainder is wrong (110) and the message is corrupted. The receiver knows that this error has occurred because the two remainders don’t match. The receiver is expecting 100 appended to the message and receives 110 appended instead, recognizing an error has occurred.

**24.** Suppose you are designing a sliding window protocol for a 1-Mbps point-to-point link to the moon, which has a one-way latency of 1.25 seconds. Assuming that each frame carries 1 KB of data, what is the minimum number of bits you need for the sequence number?

Bandwidth delay = RTT \* bandwidth = (2\*1.25s) \* (1Mbps) = 2.5Mb

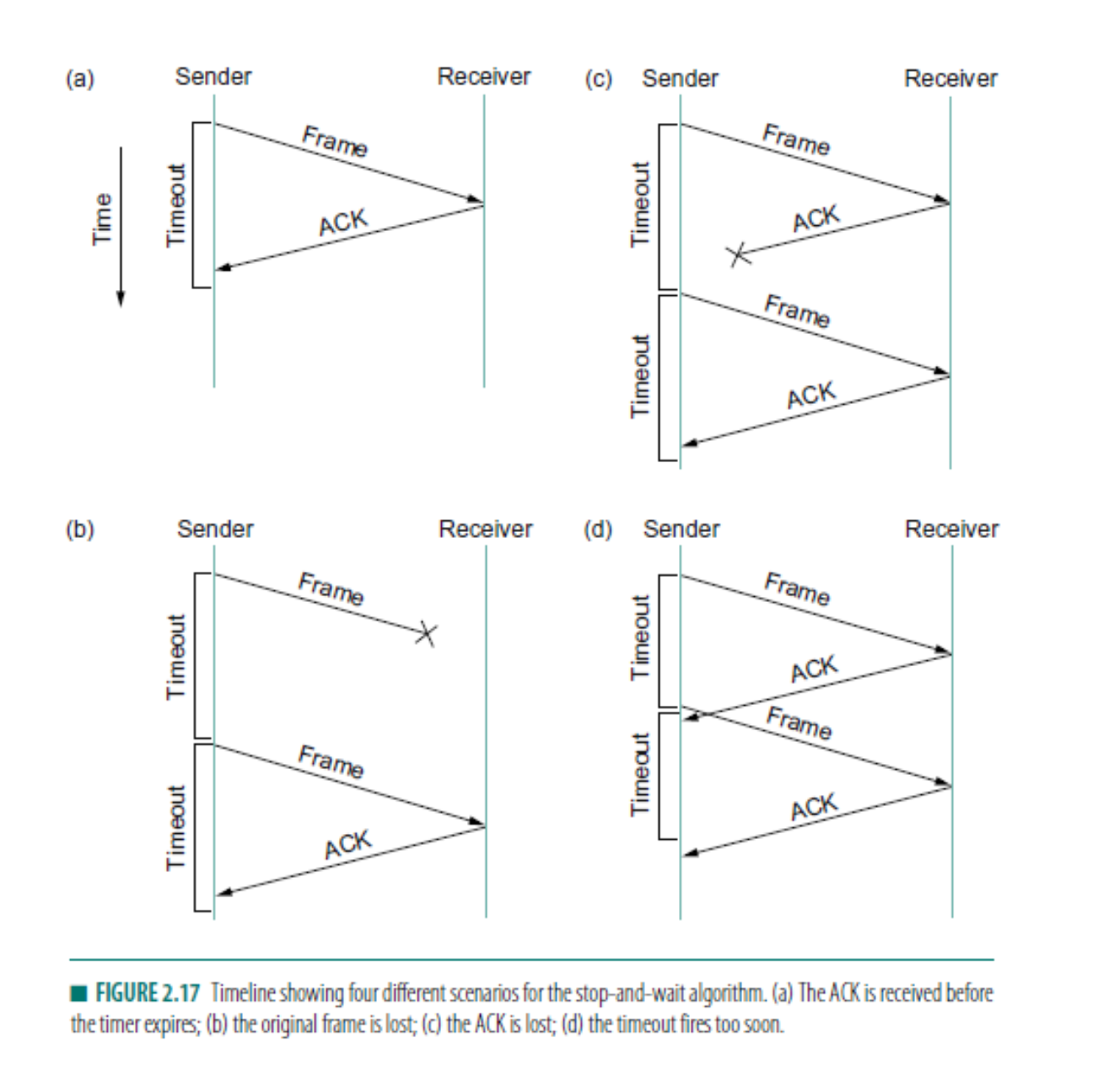
2.5Mb/ 1 KB = 2.5Mb / 8000 bits per frame = 312.5 frames

312.5 frames \* 2 = 625 maximum sequence size

minimum number of bits = 10

**27.** Implicit in the stop-and-wait scenarios of Figure 2.17 is the notion that the receiver will retransmit its ACK immediately on receipt of the duplicate data frame. Suppose instead that the receiver keeps its own timer and retransmits its ACK only after the next expected frame has not arrived within the timeout interval. Draw timelines illustrating the scenarios in Figure 2.17(b) to (d); assume the receiver’s timeout value is twice the sender’s. Also redraw (c) assuming the receiver’s timeout value is half the sender’s.

BEFORE:



Answer on attached paper

**31.** Draw a timeline diagram for the sliding window algorithm with SWS = RWS = 3 frames, for the following two situations. Use a timeout interval of about 2×RTT.

**(a)** Frame 4 is lost.

**(b)** Frames 4 to 6 are lost.

Answer on attached paper

**35.** Suppose that we run the sliding window algorithm with SWS = 5 and RWS = 3, and no out-of-order arrivals.

**(a)** Find the smallest value for MaxSeqNum. You may assume that it suffices to find the smallest MaxSeqNum such that if DATA[MaxSeqNum] is in the receive window, then DATA[0] can no longer arrive.

MaxSeqNum = 8

This is true because if DATA[8] is in the receive window, then it is impossible for DATA[0] to arrive at the receiver. The earliest receive window is DATA[6-8], and when this occurs ACK[6] is received. This means that DATA[5] was delivered successfully. SWS=5, all DATA[0]’s data was send before DATA[5]. Because data must travel in order, and not be jumbled because this would cause an error, DATA[0] cannot arrive

**(b)** Give an example showing that MaxSeqNum−1 is not sufficient.

If we try MaxSeqNum= 7 then the receiver should be expecting to receive DATA[7] and DATA[0] can still arrive. The receiver can’t tell the difference between 7 and 0 because of the modulus operation on these.

7 mod 7=0 and 0 mod 7=0

Because of this, the receiver cannot tell this difference. It will send DATA[0-4] and these will all arrive successfully. Sender sends ACK[5] but this is slow and the receive window now sends DATA[5-7]. This causes a timeout and the sender resends DATA[0], which it reads as DATA[7] because 0 mod 7=0 == 7 mod 7.

**(c)** State a general rule for the minimum MaxSeqNum in terms of SWS and RW

MaxSeqNum >= SWS + RWS