Nima Amir Dastmalchi (505320372)

**Problem 1**

* 1. Consider input of all 0s of length n. In this case, the length of the coded data is also n. This is because each 0 byte will be a block of length 1 that does not contain any other data. For example:

Input: 000000000 (each 0 is 1 byte)

Coded data: 111111111 (each 1 is 1 byte)

In this case, each block of 1 will be decoded as 0.

In general, the coded data will be almost the same length as the input data. If the data contains enough zeroes to create a coded blocks of length less than 254 bytes, then the output will be the exact same length. This is because any data of format D0 (where D consists of non-zero bytes) is encoded as XD, where x is a single byte indicating the size of D. Note that size of D0 is equal to that of XD. On the other hand, blocks of size 254 bytes will be encoded as blocks of 255 bytes (1 byte for the size (255) and 254 bytes for the data). Thus, some blocks may contain an extra byte of data.

* 1. The worst-case is when the data does not contain the byte 0. Consider large packet D of length L >> 254 such that D does not contain any 0 bytes. In this case, an extra byte is necessary (for the length byte) for each block. There will be L % 254 non-data bytes. Thus, the overhead is (L % 254) / 254.

If D has length L << 254, the worst-case is when D has no trailing 0s.

**Problem 2**

* 1. Shift M by 4 bits (G is 5 bits, so shift by 4 bits)

Calculate the remainder of shifted M divided by G

Remainder 0101?

M + CRC =1001010101 = x^9 + x^6 + x^4 + x^2 + 1 (should be divisible by G)

* 1. Add G to M + CRC to get a new M + CRC divisible by G (undetected error)
  2. This generator detects all 1-bit errors. To have an undetected error, we need to add the error polynomial to the message, where k is another non-zero polynomial. Because , will have at least 3 terms. A 1-bit error requires the addition of a 1-term polynomial to the message. Therefore, no 1-bit error will be undetected by this generator.
  3. If the generator has a factor of , then the CRC can catch all odd bit errors. Because divided by has a non-zero remainder of , is not a factor of . Thus, this generator does not detect all odd bit errors.
  4. Any polynomial represented by a bit string between and can cause an undetected burst of length 10. This range contains possible polynomials.
  5. For any 4-degree polynomial
  6. x^9 + … + 1

Find all error polynomials B of form (x^(k-9-1) + … + 1) such that G \* B = x^k-1 + … +1

* 1. B : num possibilities of (x^(L – k – 1) + … + 1) = 2\*\*(L – k – 2)

Each B corresponds to a error polynomial => undetected error possibilities = 2\*\*(L – k – 2) / 2\*\*(L – 2) = 1/2^k

**Problem 3**

* The STATUS is needed because if the last packet in the sequence is lost, then the receiver will not be able to detect an error. If the receiver crashes (no longer works), no NACK will be sent and the sender thinks that the receiver is receiving the messages. Also, if the NACK gets lost, then the sender will not retransmit. Finally, all packets could be lost and no NACK will be sent by the receiver.
  + Problem with NACK: what if receiver gets packets out of order and falsely detects a packet drop and sends a NACK.
  + Fix: Add STATUS
* Because the STATUS message acts as a periodic acknowledgement to data.
* The timer can be stopped when a STATUS message is sent, sent data is acknowledged, and no more data is sent by the sender.