Homework for Chapter 3

1. You receive the following data fragment: `0110 0111 1100 1111 0111 1101`. You know that the protocol uses bit stuffing. Show the data after destuffing.

   Solution:

Before:	`0110 0111 1100 1111 0111 1101`
After:	`0110 0111 11 0 1111 0111 11 1`

2. An 8 bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?

   Solution:

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
		1		0	1	0		1	1	1	1

$$\begin{array}{r|cccc}
P_3: & 0 & 0 & 1 & 1 \\
P_6: & 0 & 1 & 1 & 0 \\
P_9: & 1 & 0 & 0 & 1 \\
P_{10}: & 1 & 0 & 1 & 0 \\
P_{11}: & 1 & 0 & 1 & 1 \\
P_{12}: & 1 & 1 & 0 & 0 \\
\hline
\text{XOR}: & 0 & 0 & 0 & 1
\end{array}$$

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
1	0	1	0	0	1	0	0	1	1	1	1

3. A 12 bit odd-parity Hamming code whose hexadecimal value is 0xB4D arrives at a receiver. What was the original value in hexadecimal? Assume that not more than 1 bit is in error.

   Solution:

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
1	0	1	1	0	1	0	0	1	1	0	1

$$\begin{array}{r|cccc}
P_3: & 0 & 0 & 1 & 1 \\
P_6: & 0 & 1 & 1 & 0 \\
P_9: & 1 & 0 & 0 & 1 \\
P_{10}: & 1 & 0 & 1 & 0 \\
P_{12}: & 1 & 1 & 0 & 0 \\
\hline
\text{XOR}: & 1 & 0 & 1 & 0 \\
\hline
\text{NOT}: & 0 & 1 & 0 & 1
\end{array}$$

No error!!!

4. Hamming codes have a distance of three and can be used to correct a single error or detect a double error. Can they be used to do both at the same time? Explain why or why not.

   In general, if the Hamming distance is n, how many errors can be corrected? How many errors can be detected?

   Solution:

   Hamming codes have a minimum distance of 3, which means that the decoder can detect and correct a single error, *but it cannot distinguish a double bit error of some codeword from a single bit error of a different codeword*.

   Thus, some double-bit errors will be incorrectly decoded as if they were single bit errors and therefore go undetected, unless no correction is attempted.

With Hamming distance n, up to $n-1$ bits errors can be detected and up to $(n-1)/2$ bit errors can be corrected, but not both at the same time.

5. Give a formula for the lower limit on the number of redundant bits r, that need to be added to a message m, to correct all single and double errors.

   Solution:

$$\left(1+m+r+\binom{m+r}{2}\right)2^m \leq 2^{m+r} \Rightarrow \left(1+m+r+\binom{m+r}{2}\right) \leq 2^r.$$

6. Suppose that a message 1001 1100 1010 0011 is transmitted using the Internet Checksum (4 bit word). What is the value of the checksum?

   Solution:

$$\begin{aligned} 0011+1010 &= 1101 \\ 1101+1100 &= 1001+1=1010 \\ 1010+1001 &= 0011+1=0100. \end{aligned}$$

7. A bit stream 10011101 is transmitted using the standard CRC method described in the text. The generator polynomial is x^3+1. Show the actual bit string transmitted. Suppose that the third bit from the left is inverted during transmission. Show that this error is detected at the receiver's end. Give an example of bit errors in the bit string transmitted that will not be detected by the receiver.

   Solution:

   10011101100.

   10111101100 CRC Mod 1001 = 100.

   If the transmitted bit stream is converted to any multiple of 1001, the error will not be detected.

   A trivial example is if all ones in the bit stream are inverted to zeros.

8. Data link protocols always put the CRC in a trailer rather than in a header. Why?

   Solution:

   The CRC is computed during transmission and appended to the output stream as soon as the last bit goes out onto the wire.

   If the CRC were in the header, it would be necessary to make a pass over the frame to compute the CRC before transmitting. This would require each byte to be handled twice: once for checksumming and once for transmitting.

   Using the trailer cuts the work in half.

9. A stop-and-wait protocol achieves 25% bandwidth efficiency using 900 bit frames over a channel with a one-way propagation delay of 50 msec. What is the bandwidth of this channel in bits per second?

   Solution:

$$\frac{1}{2BD/L+1} = \text{Effciency } p \Rightarrow B = \frac{\frac{1}{p}-1}{2D}L = \frac{\frac{1}{0.25}-1}{2\times 50\times 10^{-3}}900 = 27\times 10^3.$$

10. A 3000-km-long T1 trunk is used to transmit 64 byte frames using protocol 5. If the propagation speed is 6 µsec/km, how many bits should the sequence numbers be?

    Solution:

    Let T be the time from sending the 64 byte frame to receiving its acknowledgement.
    Let T_1 be the transmission time for the 64 byte frame.
    Let T_2 be the propagation from the sender to the receiver to the sender.

$$T = T_1 + T_2 = \frac{64 \times 8}{1.546 \times 10^6} + 2 \times 3000 \times 6 \times 10^{-6} = \left(\frac{1}{3} + 36\right) \times 10^{-3}.$$

$$w = \frac{T}{T_1} = \frac{\left(\frac{1}{3} + 36\right) \times 10^{-3}}{\frac{1}{3} \times 10^{-3}} = 109 \leq 2^7.$$

11. Imagine a sliding window protocol using so many bits for sequence numbers that wraparound never occurs. What relations must hold among the four window edges and the window size, which is constant and the same for both the sender and the receiver?

Solution:

Let the sender's window be (S_l, S_u) and the receiver's be (R_l, R_u).

Let the window size be W.

The relations that must hold are:

$$0 \leq S_u - S_l + 1 \leq W,$$

$$R_u - R_l + 1 = w,$$

$$S_l \leq R_l \leq S_u + 1.$$

12. In protocol 6, when a data frame arrives, a check is made to see if the sequence number differs from the one expected and *no_nak* is true. If both conditions hold, a NAK is sent. Otherwise, the auxiliary timer is started. Suppose that the else clause were omitted. Would this change affect the protocol's correctness?

Solution:

Yes. It might lead to deadlock.

Suppose that a batch of frames arrived correctly and was accepted. The receiver would advance its window. Now suppose that all the acknowledgements were lost. The sender would eventually time out and send the first frame again. The receiver would then send a NAK.

If this packet were lost, from that point on, the sender would keep timing out and sending a frame that had already been accepted, but the receiver would just ignore it. Setting the auxiliary timer results in a correct acknowledgement being sent back eventually instead, which resynchronizes.

13. Frames of 1000 bits are sent over a 1 Mbps channel using a geostationary satellite whose propagation time from the earth is 270 msec. Acknowledgements are always piggybacked onto data frames. The headers are very short. Three-bit sequence numbers are used. What is the maximum achievable channel utilization for

1. Stop-and-wait?
2. Protocol 5?
3. Protocol 6?

Solution:

$$w = \frac{2BD}{L} + 1 = \frac{2 \times 10^6 \times 270 \times 10^{-3}}{10^3} + 1 = 541.$$

Stop-and-wait: $\frac{1}{w} = \frac{1}{541} = 0.185\%$.

Protocol 5: $\frac{7}{w} = \frac{7}{541} = 1.294\%$.

Protocol 6: $\frac{4}{w} = \frac{4}{541} = 0.739\%$.

14. Give at least one reason why PPP uses byte stuffing instead of bit stuffing to prevent accidental flag bytes within the payload from causing confusion.

    Solution:

    PPP was clearly designed to be implemented in software, not in hardware as bit-stuffing protocols such as HDLC nearly always are.

    With a software implementation, working entirely with bytes is much simpler than working with individual bits.

    In addition, PPP was designed to be used with modems, and modems accept and transmit data in units of 1 byte, not 1 bit.

15. What is the minimum overhead to send an IP packet using PPP? Count only the overhead introduced by PPP itself, not the IP header overhead. What is the maximum overhead?

    Solution:

    At its smallest, each frame has 2 flag bytes, 1 protocol byte, and 2 checksum bytes, for a total of 5 overhead bytes per frame.

    For maximum overhead, 2 flag bytes, 1 byte each for address and control, 2 bytes for protocol and 4 bytes for checksum. This totals to 10 overhead bytes.