Solutions to Assignment 4

Question 1

The following layers would implement the following functions:

(a) adding HDLC framing: layer 2 (link layer).

(b) specifying the protocol used by the payload: layer 2 and layer 3 (and also any higher layer that is able to carry more than one type of protocol in the payload must indicate the protocol in the payload).

(c) low-pass filtering: layer 1 (physical).

(d) addressing using an address with world-wide scope: layer 3 (network).

(e) generating/checking a CRC: typically included in layer 2 (link) but can be included at higher layers also for robustness.

(f) adding addressing using an address that is only valid for devices within a building: layer 2 (data link layer, logical link control sub-layer).

(g) amplification: layer 1 (physical).

Question 2

```
55:55:55:55:55:55:55:d5 - preamble
ff:ff:ff:ff:ff:ff       - destination address
01:02:03:04:05:06       - source address
08:00                   - type field for IPv4
```

Question 3

The large number of identical broadcast frames indicate that broadcast frames are being retransmitted in a loop. This is normally prevented by using the Spanning Tree Protocol (STP) which disables specific switch ports so as to create an (optimum) spanning tree. The STP feature was probably disabled on the reconfigured router and needs to be set back on.

Question 4

RFC 1901 is a brief introduction to SNMPv2. SNMP is a protocol used to manage (monitor and configure) networking equipment. RFC 1901 defines basic terminology (agents, management stations, MIBs, and community), how MIBs are transported between agents and management stations and the purpose of associating messages with a "community."

Question 5

The Ethernet packet can be parsed as follows (values in hex unless stated otherwise):

```
00 1d 60 9f 21 94 - destination adddress
00 1f 16 20 5d e5 - source address
08 00             - type field (IPv4)
45 00 00 34       - five 32-bit IP header words
66 98 40 00
80 06 00 00
c0 a8 03 23
89 52 10 3d
                  - IP packet payload:
      cf ec 00 50 5a a4 2f 06 00 00 00 00 80 02
20 00 8a 45 00 00 02 04 05 b4 01 03 03 02 01 01
04 02
```

(a) As shown above, the source Ethernet address is `00 1f 16 20 5d e5` and the destination Ethernet address is `00 1d 60 9f 21 94`.

(b) As shown above, the value of the length/type field is `08 00` (the IPv4 Ethertype value).

(c) The length of the IP packet can be obtained from the third and fourth bytes of the IP header: `00 34` which is 52 (decimal) bytes: 20 bytes for the header and 32 bytes for the IP payload.

(d) The protocol field in the IP header is contained in the tenth byte in the header (06) which is TCP (Transmission Control Protocol).

(e) The 16-bit IP header checksum follows the protocol and in this example has a value of `00 00`. This is most likely because the checksum computation is being "offloaded" to the network interface card.

Question 6

The most common convolutional encoder:

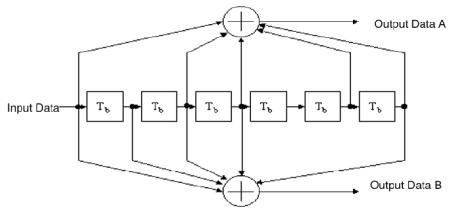

Figure 18-8—Convolutional encoder (k = 7)

computes output A as the xor of bits 6, 5, 3 2, and 0 (numbering from oldest bit to newest) and output B as the xor of bits 6, 3, 2, 1, 0. If the initial encoder state is all zeros then as the bits are shifted in we obtain the following outputs (computed using a spreadsheet and the mod(·,2) function on a sum):

0	1	2	3	4	5	6	A	B
1	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0	1
0	0	1	0	0	0	0	1	1
1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	1	0

So the output will be: 1,1, 0,1, 1,1, 0,0, 1,0.

Unfortunately, there was an error in the example which led some students to believe that the bits should have been input in reversed order (as if the input sequence was a binary number). In that case the computation would be:

1	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0
0	0	1	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0

and the output would have been: 1,0, 1,1, 0,1, 0,0, 1,0.

Question 7

An (8,2) block code has 8 bits per codeword of which 2 are data bits. With 2 data bits there must be $2^2 = 4$ codewords and for a minimum distance of 3 all pairs must differ by at least 3 bits.

Many codes are possible, but a simple way to design such a code would be to repeat a code with a minimum distance of 1 three times. For example, a 2-bit code composed of codewords 00 01 10 and 11 has a minimum distance of 1 (e.g. between 01 and 11 or 01 and 00). By repeating each codeword four times we can obtain a codeword length of 8 and a minimum distance of 4. In this case the codewords will be: 00000000, 01010101, 10101010 and 11111111.

There are in fact only $C(n,2) = C(4,2) = \frac{4!}{2(4-2)!} = \frac{24}{4} = 6$ pairs of codewords and thus 6 nonzero distances. The distance calculations can be summarized in a table:

	0000 0000	0101 0101	1010 1010	1111 1111
0000 0000	0	4	4	8
0101 0101		0	8	4
1010 1010			0	4
1111 1111				0

For example, if we transmit 0101 0101 but there is an error in the first bit so that we receive 1101 0101 then the distances to the other codewords (in order) will be: 5, 1, 7 and 3 so we would select the second codeword as the correct codeword and thus "correct" the error in the first bit.

This type of code is called a repetition code. Although very simple to design and implement, it is not widely used because many other codes have higher code rates for the same minimum distance.

For example, a Hamming (7,4) code has a minimum distance of 3 and a rate of $\frac{4}{7} \approx 0.57$ while a repetition code with the same minimum distance of 3 (e.g. a (12,4) code) has a rate of $\frac{1}{3} \approx 0.33$. Thus a Hamming code has the same error-correction performance as a 3-repetition code while transmitting data at almost twice the rate.