Homework no. 1

Submission by

- Alexander Shender 328626114
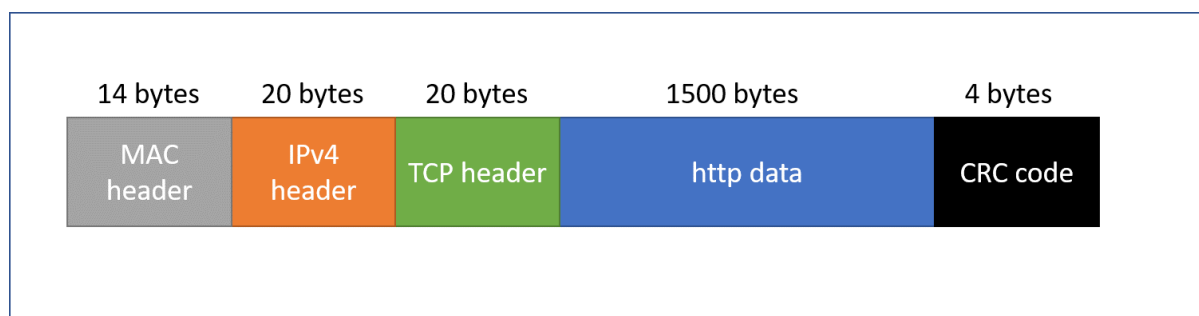- Netanel Rotschild 204937841

Question #1

(a)

The following chart shows the structure of the complete frame, where for each layer the according protocol is mentioned and the bytes number

Layer	Protocol	Header size
5	http	-
4	TCP	20
3	IPv4	20
2	MAC + CRC	14 + 4
1	PHY	-

The following diagram shows it explicitly:

Thus, total size of the frame that the PHY layer receives is 1558 bytes.

(b)

For each of the paths, the following factors define the total amount of time required for the complete transaction:

1. The transmission time
2. The propagation time
3. The parsing - reparsing time

We calculate each one separately.

1. The transmission time : T_1

As calculated, the frame has a total length of 1558 bytes = 12464 bits

Transmission time is calculated:

```
time = data size /  transmission speed
```

- Cellphone <-> WiFI access point : $t = 12464/(20 * 10^6) = 0.6232[ms]$
- WiFI access point <-> R1 : $t = 12464/(50 * 10^6) = 0.24928[ms]$
- R1 <-> R2 : $t = 12464/(10 * 10^6) = 1.2464[ms]$
- R2 <-> R4 : $t = 12464/(9 * 10^6) = 1.3849[ms]$
- R1 <-> R3 : $t = 12464/(2 * 10^6) = 6.232[ms]$
- R3 <-> R4 : $t = 12464/(2 * 10^6) = 6.232[ms]$
- R4 <-> Youtube server : $t = 12464/(100 * 10^6) = 0.12464[ms]$

So, we can calculate the total transmission time for each path.

$T_1(path1) = 0.6232 + 0.24928 + 1.2464 + 1.3849 + 0.12464 = 3.62842[ms] = 3.62842 * 10^{-3}[s]$

$T_1(path2) = 0.6232 + 0.24928 + 6.232 + 6.232 + 0.12464 = 13.46112[ms] = 13.46112 * 10^{-3}[s]$

2. The propagation time : T_2

Those times are negligible:

- Cellphone <-> WiFI access point
- WiFI access point <-> R1
- R4 <-> Youtube server

For others:

Speed of transmission = $(2/3) * (3 * 10^8) = 2 * 10^8[m/s]$

- R1 <-> R2 : $t = 3000/(2 * 10^8) = 15[us] = 15 * 10^{-6}[s]$
- R2 <-> R4 : $t = 10000/(2 * 10^8) = 50[us] = 50 * 10^{-6}[s]$

- R1 <-> R3 : $t = 6000/(2 * 10^8) = 30[us] = 30 * 10^{-6}[s]$
- R3 <-> R4 : $t = 5000/(2 * 10^8) = 25[us] = 25 * 10^{-6}[s]$

Thus,

$$T_2(path1) = (15 + 50) * 10^{-6} = 65 * 10^{-6}[s]$$

$$T_2(path2) = (30 + 25) * 10^{-6} = 55 * 10^{-6}[s]$$

3. The parsing - reparsing time : T_3

For each on the paths, this time is equal.

Assumptions:

- The Wi-Fi Access Point, R1, R2, R3, R3 only deal with the first 3 layers and do not parse the Layer 4 (Transpot layer) and Layer 5 (application)
- The Youtube serve does re-parsing of layer 4 & 5 (additionally to 1,2,3)
- The cellphone does parsing of layer 4 and 5 (additionally to 1,2,3)
- Each router does parsing and re-parsing of layer 2 and 3

So we get the total contribution of parsing - reparsing:

$$T_3 = T_{youtube} + T_R * 2 + T_{access\ point} + T_{cellphone}$$
$$T_3 = (2 * 3) + (2 * 1 + 2 * 2) * 2 + (2 * 1 + 2 * 2) + (1 * 3)[us] = 6 + 12 + 6 + 3 = 27[us] = 27 * 10^{-6}[s]$$

TOTAL

Thus, in total we obtain the following times:

$$Time(path_1) = T_1(path1) + T_2(path1) + T_3 = 3.62842 * 10^{-3} + 65 * 10^{-6} + 27 * 10^{-6}[s] = 3.72042 * 10^{-3}[s]$$

$$Time(path_2) = T_1(path2) + T_2(path2) + T_3 = 13.46112 * 10^{-3} + 55 * 10^{-6} + 27 * 10^{-6}[s] = 13.54312 * 10^{-3}[s]$$

So, path 1 is faster (through R_2). We can observe that the most decisive factor was the transmission speed. (its time is of an order higher than those of the propagation and parsing)

(c)

Now additional parsing and re-parsing of layer 4 header is added at the Wi-Fi access point. Thus, the total time for each path is increased by (1 + 2) [us] = 3 [us]

$$Time(path\ 1\ new) = 3.72042 * 10^{-3} + 3 * 10^{-6}[s] = 3.72072 * 10^{-3}[s]$$

$$Time(path\ 2\ new) = 13.54312 * 10^{-3} + 3 * 10^{-6}[s] = 13.54342 * 10^{-3}[s]$$

Question #2

(a)

Message length is 1 byte = 4 bits
To ensure we can fix 1 error, the Hamming distance has to be equal at least 3.

Using the equation from the lecture:

$$(m+r+1)\ 2^m \leq 2^{m+r}$$

where:

- m = 4

we obtain that equation hold if:

- r = 3

Thus, we have to add 3 Control bits to each frame, making the total message length equal 7.
The message will look in the following way, where C - control bits, D - data bits

1	2	3	4	5	6	7
C1	C2	D1	C3	D2	D3	D4

(b)

First, decoding HEX data to binary representation:
each HEX digit has 4 bits of data (range of 0x0 - 0xF, where 0xF = 1111)

0xA09 = 1010 0000 1001

digits at places 1, 2, 4, 8 are control digits. We have to check the parity of all the bits those bits are responsible for and verify.

Again, with the help of the lecture:

digit number	protects the follow				
1	1	3	5	7	9
2	2	3	6	7	10
4	4	5	6	7	12
8	8	9	10	11	12

Parity for each of those digits:

- 1 : even (3, 9) -> parity = 0 : INCORRECT
- 2 : uneven (3) -> parity = 1 : INCORRECT
- 4 : uneven (12) -> parity = 1 : INCORRECT
- 8 : even (9, 12) -> parity = 0 : CORRECT

As we can see, the parity is incorrect for Control bits 1, 2, 4. Thus, if we only have 1 error max. , the incorrect bit is 7.

And the correct message is:

1010 0010 1001 = 0xA29

The original message that will be passed further then consists without the control bits and is:

1001 1001 = 0x99

(c)

Now we require a full table of the control bits:

digit number	protects the following digits							
1	1	3	5	7	9	11	13	15
2	2	3	6	7	10	11	14	15
4	4	5	6	7	12	13	14	15
8	8	9	10	11	12	13	14	15

a. 1111 1110 0110 000

Parity for each of control digits:

- 1 : even (3, 5, 7, 11) -> parity = 0 : INCORRECT
- 2 : uneven (3, 6, 7, 10, 11) -> parity = 1 : CORRECT
- 4 : uneven (5, 6, 7) -> parity = 1 : CORRECT

- 8 : even (10 ,11) -> parity = 0 : CORRECT

The only incorrect control bit is '1', thus bit 1 is flipped.

Correct word:

0111 1110 0110 000

b. 1111 1111 1111 111

All the control bits are responsible on 7 Data bits. All those bits are 1, making an uneven number of bits. Thus, the control bits should be Parity = 1 to make for legit word. This is what we see. So the word has no errors!

c. 1001 1010 1001 111

Parity for each of control digits:

- 1 : uneven (5, 7, 9, 13, 15) -> parity = 1 : CORRECT
- 2 : uneven (7, 14, 15) -> parity = 1 : INCORRECT
- 4 : even (5, 7, 12, 13, 14, 15) -> parity = 0 : INCORRECT
- 8 : uneven (9, 12, 13, 14, 15) -> parity = 1 : INCORRECT

Bits 2, 4, 8 are incorrect. Thus, the incorrect bit in the message is 2 + 4 + 8 = 14.

Correct word:

1001 1010 1001 101