Emre KÖSE -
150130037

BLG337E: Principles of Computer Communications Homework #2 - REPORT

20.11.2016

BLG337E: Principles of Computer Communications
Homework #2 – REPORT

In this homework there is a sender_file_name.txt file which includes the data that consists 1's and 0's and these are also multiple of 7 and these are named data. Our task is firstly convert them to frames which has 16 bit that the first 8 bits are sequence of frames 0 to 255 in binary base, the function which converts decimal to binary:

```
char *decimal_to_binary(int n)  // decimal olarak artirilan sayi binary ye cevriliyor
{
   int c, d, count;
   char *pointer;

   count = 0;
   pointer = (char*)malloc(8+1);

   if ( pointer == NULL )
      exit(EXIT_FAILURE);

   for ( c = 7 ; c >= 0 ; c-- )
   {
      d = n >> c;

      if ( d & 1 )
         *(pointer+count) = 1 + '0';
      else
         *(pointer+count) = 0 + '0';

      count++;
   }
   *(pointer+count) = '\0';

   return  pointer;
```

and second 8 ones consist 7 bit data which comes from sender_file_name

and the last bit is parity bit which is calculated to all 16 bits as even. All these issues are handled in code in lines between 65 and 190. Before this the BER is taken from user and its range is calculated. For example, if BER is entered 0.0042, its range is 10000 and it is put on "**calculatedRange**" in program. Then to use this range BER value is converted to symbolic

value to compare with its range. In our example BER was 0.0042 and it is multiplied with "**calculatedRange**" and this is put on "**ber**" like 42 in code. To calculate the possiblity of corruption there is a random range in range of "**calculatedRange"** and a if statement in code:

```
for( i=0;i<8;i++){
  int range;
     range  =  rand() % calculatedRange;


      if((float)range < ber )
      {
          if(pointer[i]=='0'){
              printf("error occured\n");
              fputc('1' ,sent_frames_file);
              pointer[i] = '1';
          }

          else
          {   printf("error occured\n");
              fputc('0' ,sent_frames_file);
              pointer[i] = '0';
          }


      }
      else
      fputc(pointer[i] ,sent_frames_file);
```

After the possibly corrupted frames are written to "**sent_frames_file**" in sequence 0 to 255 and these are in binary base, like HW1 these frames are converted to signals with NRZ_L encoding and written to link file. This issue is handled in code lines between 195 and 243. From the link file, frames are checked according to their parity bits and possibly corrupted frames are not written to receiver_file but other frames are written. And, then again from link file frames are transported to rcvd_frames_file but there are all frames both possibly corrupted ones and possibly not corrupted ones. 'PE' indicator is added to possibly corrupted ones which are done with lines between 260 and 350 in code. **Of course this is not the ideal data transfer technique because if there is more than one possibly corrupted bit in frames, these possibilities will not point the data is reliable or not.**

Emre KÖSE -150130037