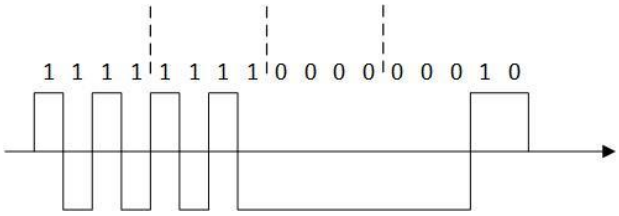
COE768 Mid-Term Solution

2018

Question 1:

a) Since the ED is 00000000, the data portion should never contain 8 consecutive zeros. To achieve this, we need to insert a "one" in the data stream after every **7 consecutive zeros**.

b) After the bit stuffing, the bit stream is 1111 1111 0000 00010.



Question 2

$x^{11} + x^8 + x^7 + x^4 \div x^4 + x^2 + 1$, the reminder is $x^3 + 1$

Question 3

a) $S_f = 15;\ S_n = 5; R_n = 8$

b) Since the number of outstanding frames at t=20msec is 3, station A can send 2 data frames with the following header contents:
- Type = Data, seq= 5, next=8.
- Type = Data, seq= 6, next=8.

c) Station A will send a NAK acknowledgement frame:
Type = NAK, next=8.

d) Station A will send the last 2 of the four data frames:
- Type=Data, seq=7, next=9.
- Type=Data, seq=8, next=9.

Question 4

a) Since $t_f = \frac{12000}{10^8} = 1.2 \times 10^{-4}$, and $t_{prop} = \frac{100}{2\times10^5} = 5 \times 10^{-4}$ therefore, $a = \frac{5\times10^{-4}}{1.2\times10^{-4}} \approx 4.2$.

Therefore, $W_s \geq 2a + 1 \approx 9.4$

For selective repeat, $N = 2^n \geq 2W_s \geq 18.8$.

Consequently, the minimum number of bits needed is 5.

b) Assume we choose the sending window size which is half of N, there is, $W_s = 16$. Thus, the datalink layer need enough buffer space to buffer all the possible out-of-sequence frames and an in-sequence frame. Therefore,

Buffer size = $W_s \times 1500$ bytes $= 24{,}000$ bytes.

Question 4

Client

```
struct PDU{
    char    type;
    char    data[BUFSIZ];
} rpdu, tpdu;
    …
    connect(sd, (struct sockaddr *) &server, alen);
    while(1){
      printf("Please enter command: T- Send message; O-over. Q-Quit \n");
      scanf("c",&cmd);                          //Read user command
      if(cmd == 'L'){                           // Download Data
            n=read(0, &tpdu.data, BUFSIZ);  //Read file name
            fd = open("rpdu.data, …);          //Open the file for read
            rpdu.type = 'L';
            write(sd, &rpdu, n+1);             //Send file name to the server
            rpdu.type = 'D';
            while((n=read(fd, &tpdu.data, BUFSIZ)) != 0)  //Read and send file data
                  write(sd, &tpdu, n+1);
            tpdu.type='F';
            write(sd, &tpdu, 1);
            close(fd);
      }
```

Server

```
while(1){
        recvfrom(sd, &rpdu, …);
        if(rpdu.type == ‘L’) {
                fd=open(rpdu.data, …);
                n = recvfrom(sd, &rpdu, …)
                while(rpdu.type == ‘D’){
                        write(fd, rpdu.data, n-1);
                        n = recvfrom(sd, &rpdu, …);
                }
                If(rpdu.type != ‘F’)
                        Printf(“protocol error\n”);
        }

}
```