Networks Lab(4)

Framing and Error Detection

Name	Sec	BN	Code
Basma Hatem Elhoseny	1	16	9202381
Mohab	2	29	9203568

Demo

Refer to the Demo attached.

Results:

Error in Payload:

```
Enter Word
hi
You entered hi
Sending...
Sender:Message Sent with Error :D at bit(18) [bit no(2)in byte (3)]
Message:hm
Header (Ch_Count):00000100
PayLoad:
                  01101000
                  01101101

Tailer (Parity):  00000101

I have Received Wrong Message :(
Final Receiver parity_check:00000100
Enter Word
```

Note:

i	105	01101001
m	109	01101101

Error in Parity:

```
Enter Word
how
You entered how
Sending...
Sender:Message Sent with Error :D at bit(33) [bit no(1)in byte (5)]
Message:how
Header (Ch_Count):00000101
PayLoad:
                    01101000
                    01101111
                    01110111

Tailer (Parity):   01110111

I have Received Wrong Message :(
Final Receiver parity_check:00000010
```

No Error with Sending:

```
Enter Word
hi
You entered hi
Sending...
Sender:Message Sent without Error :D
Message:hi
Header (Ch_Count):00000100
PayLoad:
                    01101000
                    01101001

Tailer (Parity):   00000101

I have Received Correct Message :) :hi
Final Receiver parity_check:00000000
```

Sender Code:

```
20 void Sender::initialize()
21 {
22     // TODO - Generated method body
23     //Hello
24     EV<<"Hello from Sender Node"<<endl;
25
26     //Self-Message
27     scheduleAt(simTime(),new cMessage("")); // Send Now :D
28 }
29
```

```
30 void Sender::handleMessage(cMessage *msg)
31 {
32     // TODO - Generated method body
33     if(msg->isSelfMessage()){
34         // Handle of Self Messaging
35
36         /*=============================================================================================
37         //Step (1):Takes Input From The user
38         std::string message;
39         std::cout<<"Enter Word"<<endl;
40         std::cin>>message;
41         std::cout<<"You entered "<<message<<std::endl;
42         int char_count=message.size();
```

```
        /*=============================================================================================
        //Step(2): Processing Message
        std::cout<<"Sending..."<<endl;
        std::vector<std::bitset<8> > packet(char_count+2); //each char is 1 byte

        //Calculate character count
        packet[0]=std::bitset<8>(char_count+2);//No of chars+1 for count_header +1 for parity

        //Even Parity Check
        std::bitset<8> parity_check=packet[0];


       //Pay_load
        for(int i=0;i<char_count;i++){
            //Add to pay load
            packet[i+1]=std::bitset<8>(message[i]);

            //Add to Parity
            parity_check^=packet[i+1];
        }

        //Update Tailer
        packet[char_count+1]=parity_check;
```

8bits	8*n bits	8bits
Header Ch Count	Payload	Tailer Even Priority

```
/*=============================================================================================
//Step(3) Adding Noise
double error = par("error");
bool error_msg=false;
int error_bit, byte_no,indx;
if(error>0.5){
    //set error
    cPar& msg_len_par = par("msg_len"); //update with the current message length
    msg_len_par.setIntValue(char_count+2);
    //bit to toggle
    error_bit=par("error_bit");

    byte_no=error_bit/8;
    indx=error_bit-8*byte_no;
    //Negation
    packet[byte_no][indx]=~ packet[byte_no][indx];

    error_msg=true;
}
```

```
//
simple Sender
{
    parameters:
        volatile double error = uniform(0,1); //uniform distrbution of 50% chance of error
        int msg_len @mutable= default(0) ;  //Message Length
        volatile int error_bit = int(uniform(0,msg_len*8)); //uniform distrbution of the bit to be toggled depe

    gates:
      output out;
}
```

```
/*=============================================================================================
//Step(4) Sending Message
std::string packet_str="";
std::string msg_str="";
//Char_Count
packet_str+=(char)(packet[0].to_ulong()+48);

//Pay_load
for(int i=0;i<char_count;i++){
    msg_str+= (char)packet[i+1].to_ulong();
}
packet_str+=msg_str;

//Parity
packet_str+=(char)(packet[char_count+1].to_ulong()+48);

//Set Content of the message
send(new cMessage (packet_str.c_str()),"out");
```

```
if(msg->isSelfMessage()){
    // Handle of Self Messaging


     }
 else{
    //Reply From The Receiver he has received my previous message Either Correctly or Incorrectly
    //Self Message to wait (listen) for input from the user
     scheduleAt(simTime(),new cMessage(""));
 }
```

Receiver Code:

```
19
20 void Receiver::initialize()
21 {
22     // TODO - Generated method body
23     EV<<"Hello from Receiver Node"<<endl;
24 }
25
```

```
24 }
25
26 void Receiver::handleMessage(cMessage *msg)
27 {
28     // TODO - Generated method body
29     /*=============================================================================================*/
30     //Step(1) Decompose Message
31     std::string packet=msg->getName();
32 //    int packet_size=packet.size();
33
34
35     //Char_count
36      std::bitset<8> char_count_header=std::bitset<8>(packet[0]-48);
37      int char_count= packet[0]-48;
38
39     //Pay_load
40     std::vector<std::bitset<8> > payload(char_count-2); //each char is 1 byte
41     std::string message="";
42     for(int i=1;i<char_count-1;i++){
43            //Add to pay load
44            payload[i-1]=std::bitset<8>(packet[i]);
45            message+=packet[i];
46     }
47     //Tailer
48     std::bitset<8> parity_check=std::bitset<8>(packet[char_count-1]-48);
49
```

```
/*=============================================================================
//Step(2) Check Errors
std::bitset<8> parity_check_xor=parity_check;
parity_check_xor^=char_count_header;
for(int i=0;i<payload.size();i++){
    parity_check_xor^=std::bitset<8>(payload[i]);
}

bool error=parity_check_xor.to_ulong();
if(!error){
    //no Error
    std::cout<<"I have Received Correct Message :) :"<<message<<endl;
}
else{
    std::cout<<"I have Received Wrong Message :("<<endl;
}
```

```
/*=============================================================================
//Step(4) Send A dummy Message to the receiver that you have received the message
send(new cMessage (""),"out");
```