Cairo University
Faculty of Engineering
Computer Department

Computer Networks-1
CMP405-A /CMPN405
Fall 2023

Framing and Error detection

lab4

1

Lab Objectives

- Given a packet from the network layer ,
  The data link layer should add the framing and the error detection bits to it .

- In this lab, we will implement the character count framing and the parity check bits for error detection.

Lab Objectives

- Say the payload input form network layer is 'hi'.
- The output : '00000100 01101000 01101001 00000101'

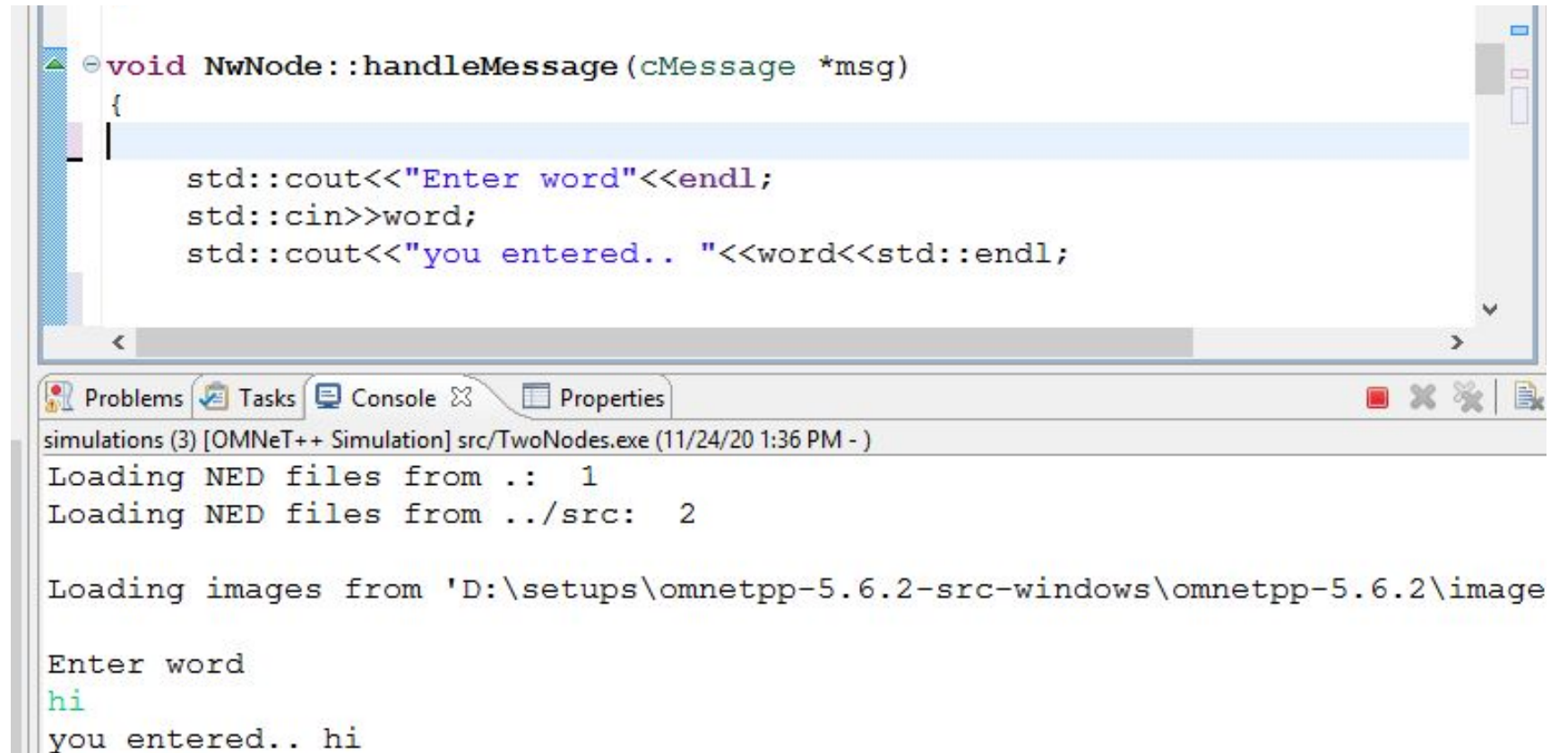| Header (Ch_count)	Payload 'h'	Payload 'i'	Parity checks (even parity)
00000100	01101000	01101001	00000101

C++ Needed functions revision
1) cin/cout/string

- You have to write 'std::' before functions that belong to the std library in the c++ files.

- You can use the cin and cout but it will out the results on the console instead of inside the simulation.

- During simulation , you may need to press step in the simulation window then return back to the editor window to access the console.

C++ Needed functions
1) cin/cout/string revision

 For example :

```
void NwNode::handleMessage(cMessage *msg)
{

    std::cout<<"Enter word"<<endl;
    std::cin>>word;
    std::cout<<"you entered.. "<<word<<std::endl;
```

Problems Tasks Console ⊠ Properties

simulations (3) [OMNeT++ Simulation] src/TwoNodes.exe (11/24/20 1:36 PM -)

```
Loading NED files from .:   1
Loading NED files from ../src:   2

Loading images from 'D:\setups\omnetpp-5.6.2-src-windows\omnetpp-5.6.2\image

Enter word
hi
you entered.. hi
```

C++ Needed functions revision
1) cin/cout/string

- Use function string.size() // gets the size of the string.

- Remember that the string is just a character array, and each character is one byte.

C++ Needed functions revision
2) bitset

- The bitset STL library is used to make operations on the bit level.
- You have to include it first inside the '.h' file.
- #include <bitset>
- In the c++ you have to write std:: before the bitset definition.
- The bitset constructor can take characters, integers, strings ,…etc.
- It has much more functionalities but we will revise what we need.

C++ Needed functions revision
2) bitset

```cpp
std::cout<<"bitset examples"<<endl;
int x= 5;
std::bitset<8> xbits (x);
std::cout<< xbits.to_string() <<endl;

char ch = 'A';
std::bitset<8> chbits (ch);
std::cout<< chbits.to_string() <<endl;
```

Problems | Tasks | Console ⊠ | Properties

simulations (3) [OMNeT++ Simulation] src/TwoNodes.exe (11/24/20 2:07 PM -)

```
bitset examples
00000101
01000001
```

C++ Needed functions revision
2) bitset

- You can do bitwise operations as well .

```cpp
std::bitset<4> bset1(9); // bset1 contains 1001
std::bitset<4> bset2(3); // bset2 contains 0011
std::cout << (bset1 & bset2) << endl; // 0010
std::cout << (~bset2) << endl; // 1100
std::cout << (bset1 ^ bset2) << endl; // 1010
```

Problems | Tasks | Console ✕ | Properties

simulations (3) [OMNeT++ Simulation] src/TwoNodes.exe (11/24/20 2:15 PM -)

```
bitset examples
0001
1100
1010
```

C++ Needed functions revision
2) bitset

- Use **bitset.to_string()** to convert the binary stream to binary string.

- Use **bitset.to_ulong()** to convert the binary stream to unsigned integer.

- Use casting like **(char) bitset.to_ulong() to convert the binary stream to the asci character.**

C++ Needed functions revision
3) vectors/iterators

- The vector STL library is used to construct easy and dynamic c++ arrays.

- The vector can store any type [int , string ,bitset, ...etc.]

- You have to include it first inside the '.h' file.

- #include <vector>

- In the c++ you have to write std:: before the vector definition

```
std::vector<int> v;
v.push_back(4);
v.push_back(2);
v.pop_back();
```

C++ Needed functions revision
3) vectors/iterators

 You can loop on any vector by iterator of the same type of the vector

```
std::vector<int> v;
v.push_back(4);
v.push_back(2);
v.pop_back();

for (std::vector<int>::iterator it = v.begin(); it != v.end(); ++it)
{
    std::cout<<*it<<endl;
}
```

 The same applies for vector of bitset (s)

std::vector<std::bitset<8>|> vec;   [take care of the space >| >]

Lab4 requirement

- Design a simple network of two nodes called "sender" and "receiver".

- The sender node takes the input string from the user "string of any length".

- The sender node, will convert the string to it's ASCI representation, add the frame header "character count", add the frame trailer "error detection byte", and send the bitstream to the receiver node.

- Before sending, the sender will have a 50% chance of introducing a single bit error to any bit in the bitstream using the uniform random distribution .

Lab4 requirement

 The sender also has to print the bitstream _before the single bit error and after it if any_, in the console in the following format.

message "hi" :

00000100

01101000

01101001

00000101

 You should also print which bit is modified.

Lab4 requirement

- The receiver node would check if there is any thing wrong in the received bitstream, if there is error, it should print an error message.

- If the message is correct, the receiver will convert the message back to its original string given from the user and print it like the following.
The original message is "hi".

- You are not to make the sender send the "original string" obviously.

Lab4 requirement helping notes

- You have to do XOR operation on all the payload characters to calculate the even parity check bits. *What is the easiest way to do this ?*

- Before sending the message back we have to convert it to c-string as message name is of type c-string, so what to do ?

  - *Convert each char to asci again using **(char)bitset.to_ulong()** .*

  - *Append the characters to one string .*

  - *Send the string as the message name.*

  - *At the receiver side, do the reverse conversion again.*

Thank You !!