Name : Rajat Minekar
Roll no : 31448, M4

Page No.
Date

Assignment - II

Title :- To write a program for error detection & correction of 7 bits ASCII codes using Hamming code or CRC. Demonstrate the packets captured traces using Wireshark packet Analyzer.

Objective :- i) To understand error detection & correction in computer networks.
ii) To familarize with tools like wireshark.

Outcome :- To implement hamming code & CRC techniques for error detection.

Concept related theory :-

In digital system, analog signals will change into digital seq.

This can cause errors while message is being delivered.

There are many types of error detection

1) Hamming Code
2) CRC. Cyclic Redundancy check.

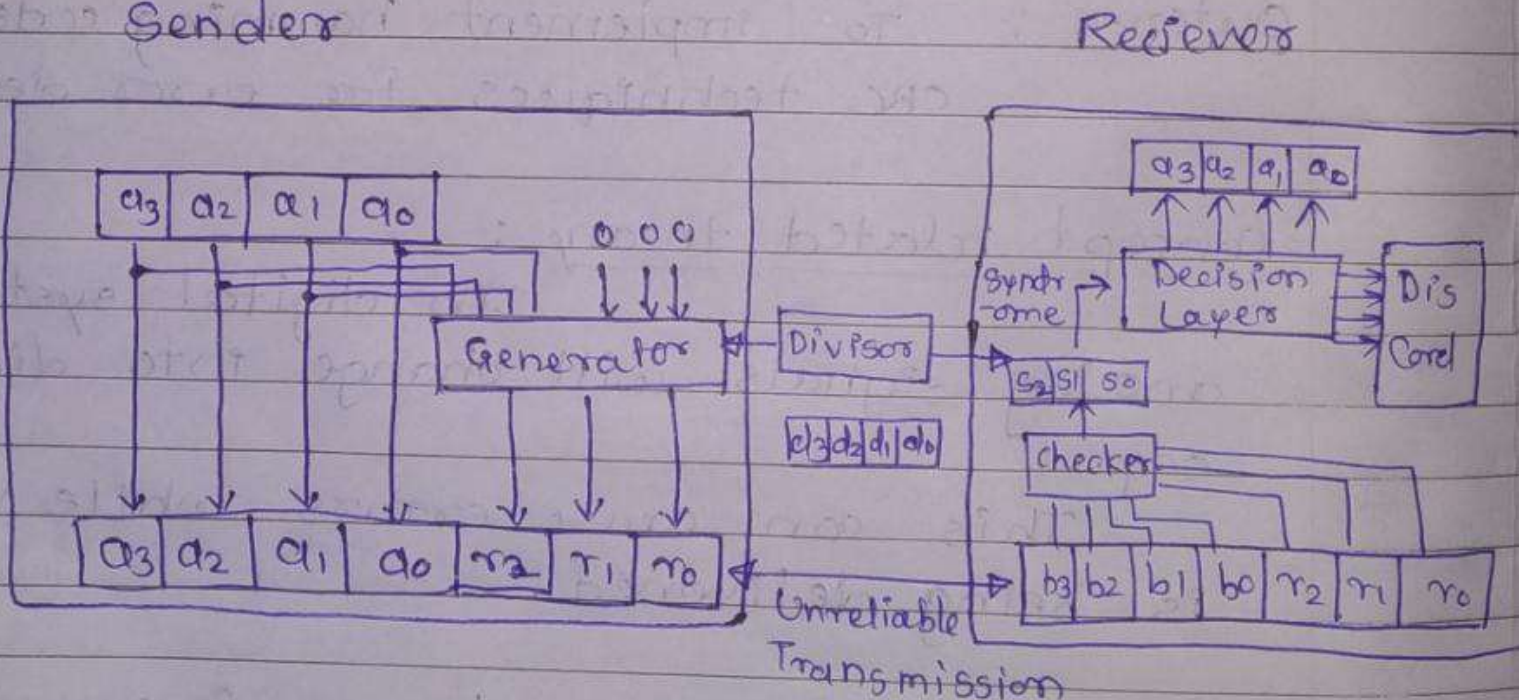
Page No.
Date

Hamming Code :-

It is a block of code that is capable of detecting upto 2 simultaneous bit errors & correcting single bit errors.

Cyclic Redundancy Check :-

- CRC is a block code commonly used to detect accidental changes in data transmitted.
- CRC includes binary division of data bits by a predefined divisor.

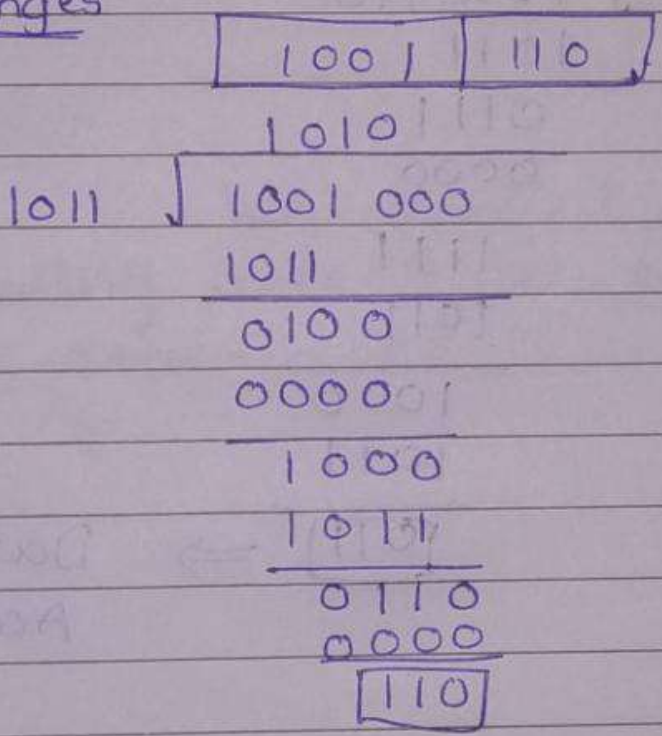


Page No.
Date

Algorithm :-

- Start.
- Take message M & Generator G.
- Using divisor by performing division, we get CRC output as Syndrome.
- Now, create disrepancies in Reciever side if wanted.
- check Syndrome throgh decision layer.
- If all 0's are obtained, there is no error.
- Else, error is detected.
- End.

changes



Page No.
Date

1001 | 110

```
          1010
1011 ) 1001110
        1011
        0101
        1011
        01011
         1011
         0000
         0000
          000
```

[000] ⇒ Dataword Accepted.
syndrome

1000 | 110

```
1011 ) 1000110
        1011
        0111
        0000
         1111
         1011
          1000
          1011
           011
```

[011] ⇒ Dataword not Accepted.

Page No.
Date

Test case :-

Input	Expected O/P	Actual O/P	Result
M = 1001 G = 1010 RM = 1001110	CRC = 110 TM = 1001110 No Error	same	Success
M = 1001 G = 1010 RM = 1000110	CRC = 110 TM = 1001110 Error in Message	Same	Success

Conclusion :-

We studied error detection using Hamming code & CRC technique through this experiment.

```
#include<iostream>
#include<string.h>
using namespace std;

#define N strlen(g)

char t[28],cs[28],g[] = "1011";
int a,e,c;


void xor()
{
 for(c = 1;c<N ; c++)
 {
  cs[c] = ((cs[c] == g[c]) ? '0' : '1');
 }
}

void crc()
{
 for(e=0;e<N;e++)
 {
  cs[e]=t[e];
 }

 do
 {
  if(cs[0] == '1')
   xor();
  for(c = 0;c<N-1;c++)
  {
   cs[c] = cs[c+1];
  }
  cs[c] = t[e++];

 }while(e<= a+N-1);

}


int main()
{
 cout<<"\n\tEnter data : ";
 cin>>t;
 cout<<"\n\t-------------------------------------";
 cout<<"\n\tGenerating polynomial : "<<g;
 a= strlen(t);
 for(e=a;e<a+N-1;e++)
 {
  t[e] = '0';
 }
 cout<<"\n\t-------------------------------------";
 cout<<"\n\tModified data is : "<<t;
 cout<<"\n\t-------------------------------------";
```

```
crc();
cout<<"\n\tChecksum is : "<<cs;
for(e=a;e<a+N-1;e++)
{
 t[e]=cs[e-a];
}
cout<<"\n\t-------------------------------------";
cout<<"\n\tFinal codeward is : "<<t;
cout<<"\n\t-------------------------------------";
cout<<"\n\tTest error detection 0(yes) 1(no) : ";
cin>>e;
if(e==0)
{
 do{
  cout<<"\n\tEnter the position where error is to be inserted : ";
  cin>>e;
 }while(e==0 || e>a+N-1);
 t[e-1] = (t[e-1] == '0') ? '1' : '0';
 cout<<"\n\t---------------------------------";
 cout<<"\n\tErroneous data : "<<t<<"\n";
}
crc();
for(e=0;(e<N-1) && (cs[e] != '1');e++)
{
 if(e<N-1)
 {
  cout<<"\n\tError Detected \n\n";
 }
 else
 {
  cout<<"\n\tNo Error Detection \n\n";
 }
}
cout<<"\n\t-------------------------------------\n\n";
return 0;


}
```


D:\TE\TE sem-1\CNL\A2\A2_C.exe
Enter data : 1001

Generatng polynomial : 1011

Modified data is : 1001000

Checksum is : 110

Final codeword is : 1001110

Test error detection 0(yes) 1(no)? : 1
No error detected

Process exited after 17.78 seconds with return value 0
Press any key to continue . . .


D:\TE\TE sem-1\CNL\A2\A2_C.exe
Enter data : 1001

Generatng polynomial : 1011

Modified data is : 1001000

Checksum is : 110

Final codeword is : 1001110

Test error detection 0(yes) 1(no)? : 0
Enter the position where error is to be inserted : 4

Erroneous data : 1000110
Error detected

Process exited after 19.65 seconds with return value 0
Press any key to continue . . .