Name	Rohit Ashiwal
Enr. No.	17114064
Dept	CSE
Batch	CS 2
Class	B. Tech. 3rd yr

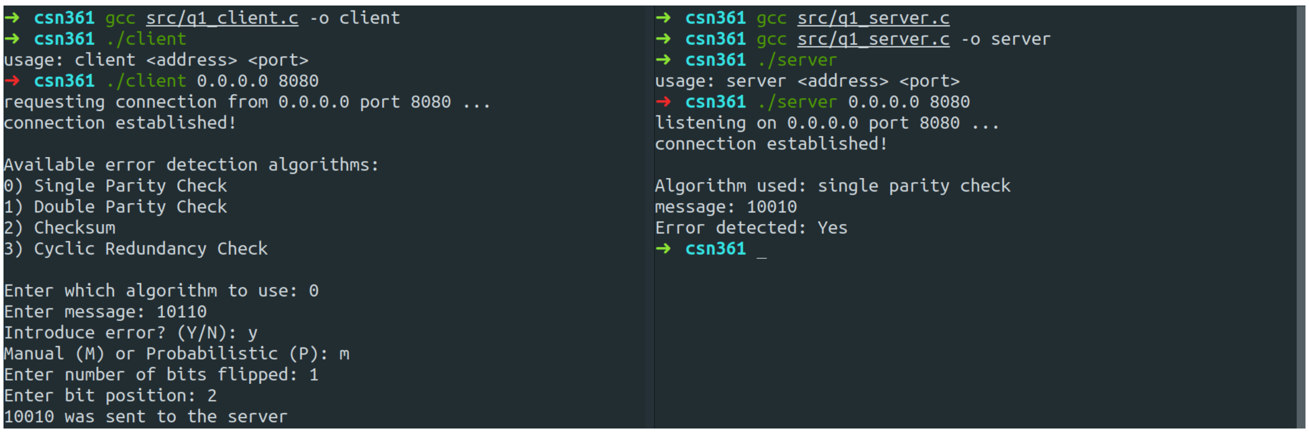
Lab Assignment 7

This assignment aims to make us familiar with the hardware and software aspects of computer networking and extracting information related to computer networking using C/C++ programs.

Problem Statement 1

Q: *Transmit a binary message (from sender to a receiver) using socket programming in C and report whether the received msg is correct or not; using the following error detection algorithms:*

```
1. Single Parity Check
2. Two-dimensional Parity Check
3. Checksum
4. Cyclic Redundancy Check (CRC)
```



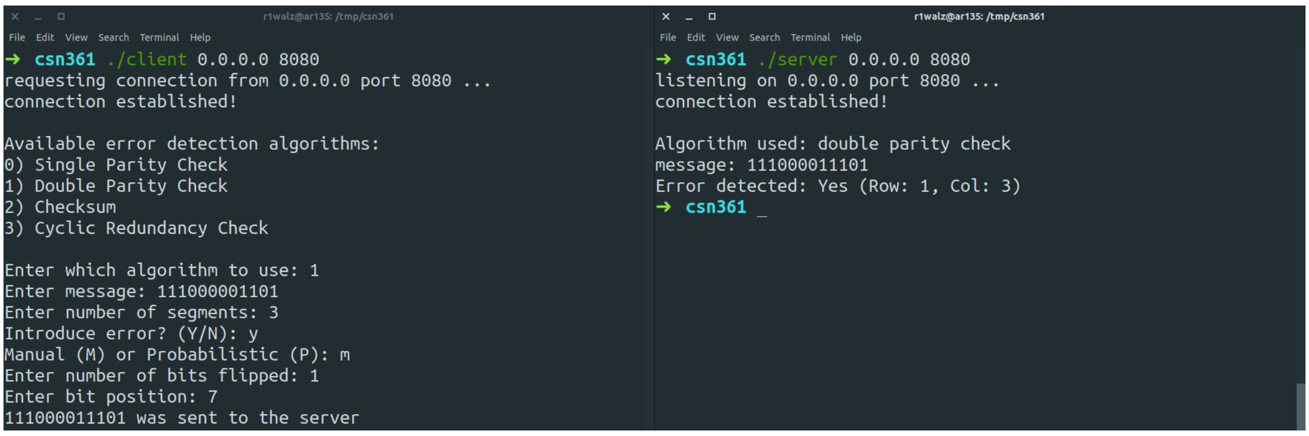
```
➜ csn361 gcc src/q1_client.c -o client
➜ csn361 ./client
usage: client <address> <port>
➜ csn361 ./client 0.0.0.0 8080
requesting connection from 0.0.0.0 port 8080 ...
connection established!

Available error detection algorithms:
0) Single Parity Check
1) Double Parity Check
2) Checksum
3) Cyclic Redundancy Check

Enter which algorithm to use: 0
Enter message: 10110
Introduce error? (Y/N): y
Manual (M) or Probabilistic (P): m
Enter number of bits flipped: 1
Enter bit position: 2
10010 was sent to the server
```

```
➜ csn361 gcc src/q1_server.c
➜ csn361 gcc src/q1_server.c -o server
➜ csn361 ./server
usage: server <address> <port>
➜ csn361 ./server 0.0.0.0 8080
listening on 0.0.0.0 port 8080 ...
connection established!

Algorithm used: single parity check
message: 10010
Error detected: Yes
➜ csn361 _
```



r1walz@ar135: /tmp/csn361
File Edit View Search Terminal Help
r1walz@ar135: /tmp/csn361
File Edit View Search Terminal Help

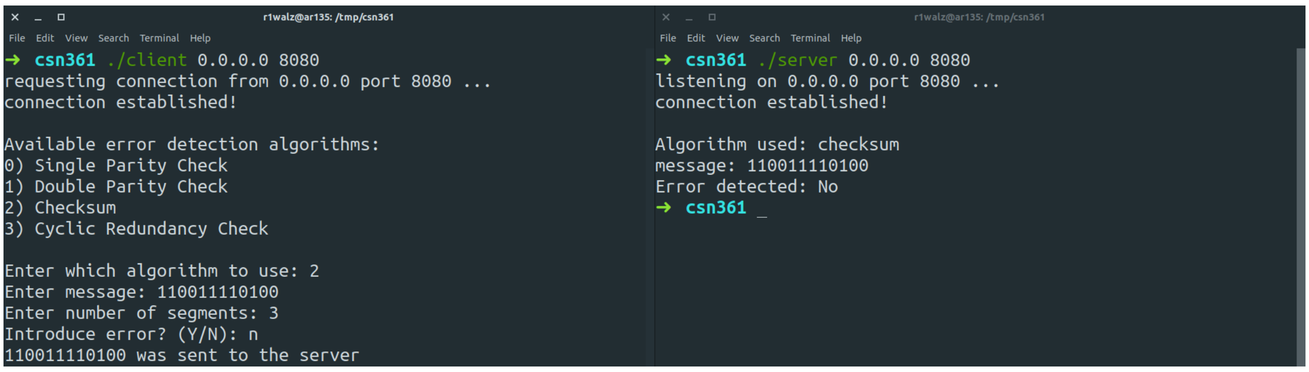
```
➜ csn361 ./client 0.0.0.0 8080
requesting connection from 0.0.0.0 port 8080 ...
connection established!

Available error detection algorithms:
0) Single Parity Check
1) Double Parity Check
2) Checksum
3) Cyclic Redundancy Check

Enter which algorithm to use: 1
Enter message: 111000001101
Enter number of segments: 3
Introduce error? (Y/N): y
Manual (M) or Probabilistic (P): m
Enter number of bits flipped: 1
Enter bit position: 7
111000011101 was sent to the server
```

```
➜ csn361 ./server 0.0.0.0 8080
listening on 0.0.0.0 port 8080 ...
connection established!

Algorithm used: double parity check
message: 111000011101
Error detected: Yes (Row: 1, Col: 3)
➜ csn361 _
```



r1walz@ar135: /tmp/csn361
File Edit View Search Terminal Help
r1walz@ar135: /tmp/csn361
File Edit View Search Terminal Help

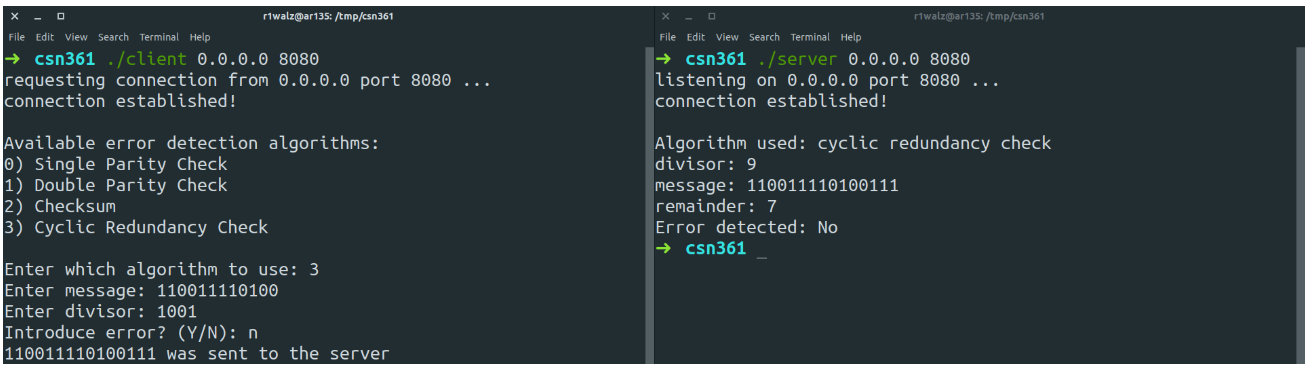
```
➜ csn361 ./client 0.0.0.0 8080
requesting connection from 0.0.0.0 port 8080 ...
connection established!

Available error detection algorithms:
0) Single Parity Check
1) Double Parity Check
2) Checksum
3) Cyclic Redundancy Check

Enter which algorithm to use: 2
Enter message: 110011110100
Enter number of segments: 3
Introduce error? (Y/N): n
110011110100 was sent to the server
```

```
➜ csn361 ./server 0.0.0.0 8080
listening on 0.0.0.0 port 8080 ...
connection established!

Algorithm used: checksum
message: 110011110100
Error detected: No
➜ csn361 _
```


r1walz@ar135: /tmp/csn361
File Edit View Search Terminal Help
r1walz@ar135: /tmp/csn361
File Edit View Search Terminal Help

```
➜ csn361 ./client 0.0.0.0 8080
requesting connection from 0.0.0.0 port 8080 ...
connection established!

Available error detection algorithms:
0) Single Parity Check
1) Double Parity Check
2) Checksum
3) Cyclic Redundancy Check

Enter which algorithm to use: 3
Enter message: 110011110100
Enter divisor: 1001
Introduce error? (Y/N): n
110011110100111 was sent to the server
```

```
➜ csn361 ./server 0.0.0.0 8080
listening on 0.0.0.0 port 8080 ...
connection established!

Algorithm used: cyclic redundancy check
divisor: 9
message: 110011110100111
remainder: 7
Error detected: No
➜ csn361 _
```

Two data structures were used:

```
1. enum algorithm: which couples all options for available error detection algorithms
2. struct pckt: which stores all information related to the data being transferred
```

All regular algorithms were implemented in their most naive form.

For adding probabilistic errors an array of random integers was generated with time(NULL) as seed to srand() and based on this array the bits of original message were flipped.

Problem Statement 2

Q: *Write a Network Simulator (NS2) code to simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion. Study the variation in number of packets dropped with the variation of the queue size in the nodes and with the variation of the bandwidth of the links.*

Network Simulator

Queue Stats

Band Stats

No special algorithms or data structures were used.

Above graphs show that with the increase in queue size / bandwidth, the number of packets drop increase with some minor fluctuations.