CS359 Computer Networks Lab 7

Name: M Maheeth Reddy	Roll No.: 1801CS31	Date: 18-Mar-2021

Cyclic Redundancy Check (CRC) : Implementation and Demonstration

---

Introduction: I have written two programs, **client.c** and **server.c**

A string is taken as input from user by the client program. The string is sent to the server in the form of ASCII and CRC remainders. The ASCII code of each character is taken, split into parts containing 4-bits each, then the corresponding CRC remainders are calculated using the CRC key = 1011, and those 7-bits are sent as a single packet to the server. The server uses the data and remainder bits to check if the data is corrupted. If the data got corrupted, it requests the client to send those bits once again. Using this, the corresponding characters are continuously concatenated to reconstruct the string being sent by the client.

Finally, the server acknowledges the client with md5sum of received string. If the md5sums at client and server sides match, the transmission was successful and Cyclic Redundancy Check's purpose is served.

Compilation: gcc client.c –o client gcc server.c –o server

Execution: ./client ./server

Sample Session: Sending the message "*H*"

Client Side

```
┌─[maheeth@maheeth-PC:~/D/netlab7]─[03:02:02  IST]
└─>$ gcc client.c -o client
┌─[maheeth@maheeth-PC:~/D/netlab7]─[03:02:14  IST]
└─>$ ./client
Enter a message: H
[+]Server socket created successfully.
[+]Connected to Server.
[+]Sending Packets.

Sending Character: "H" (ASCII 72)
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0100
Data with remainder = 0100 111
Data to be sent = 1000
Data with remainder = 1000 101
-------------
Sending Character: "\0" (ASCII 0)
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
```

… (few more lines here) …

```
[+]User message sent successfully.
MD5 client:     c1d9f50f86825a1a2302ec2449c17196
MD5 server:     c1d9f50f86825a1a2302ec2449c17196
[+]MD5 Matched
[+]Closing the connection.
```

Server Side

```
┌─[maheeth@maheeth-PC:~/D/netlab7]─[02:37:32  IST]
└─>$ gcc server.c -o server
┌─[maheeth@maheeth-PC:~/D/netlab7]─[02:38:04  IST]
└─>$ ./server
[+]Server socket created successfully.
[+]Binding successful.
[+]Listening....
Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0100 111
Data Received: 0100     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 1000 101
Data Received: 1000     Remainder after CRC checking: 000
Data was received properly

Character "H" (ASCII 72) was received from client
----------------------------------------
```

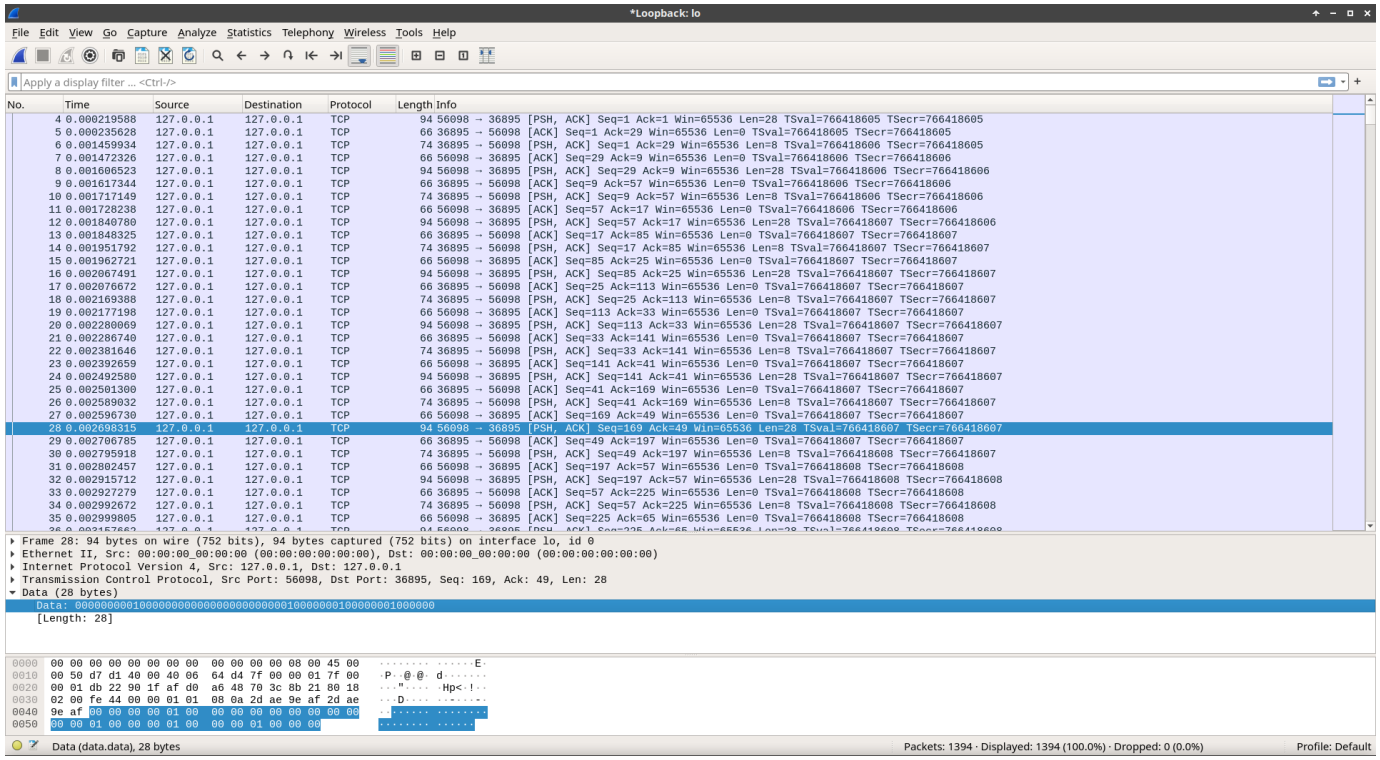
… (few more lines here) …

```
----------------------------------------
[+]Data received successfully.
Message sent by client:
        "H"
MD5:    c1d9f50f86825a1a2302ec2449c17196
[+]MD5 sent
[+]Closing the connection.
```

Wireshark Packet Capture for the following input:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Screenshot from Wireshark: Total Packets captured = **1394**



Screenshot of Client side code

```
-------------
Sending Character: "\0" (ASCII 0)
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
Data to be sent = 0000
Data with remainder = 0000 000
[+]User message sent successfully.
MD5 client:     35899082e51edf667f14477ac000cbba
MD5 server:     35899082e51edf667f14477ac000cbba
[+]MD5 Matched
[+]Closing the connection.
```

MD5sums matched => Cyclic Redundancy Check is working efficiently.

Screenshot of Server side code

```
----------------------------------------
Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Data and remainder received: 0000 000
Data Received: 0000     Remainder after CRC checking: 000
Data was received properly

Character "\0" (ASCII 0) was received from client
----------------------------------------
[+]Data received successfully.
Message sent by client:
        "Lorem ipsum dolor sit amet, consectetur adipiscing elit."
MD5:    35899082e51edf667f14477ac000cbba
[+]MD5 sent
[+]Closing the connection.
```

----------x----------------x---------