**Operating Systems Project 3**

**Client Server Encode Decode**

**Error Detection and Correction  
CRC and Hamming**

**Name: Vamsi Krishna Changala**

**ID: 001573653**

**System Documentation**

**High level Data flow diagram**

**Client:**

Diagram

Description automatically generated

**Server:**

Diagram, engineering drawing

Description automatically generated

**Description & Implementation Details:**

The "physicallayer.c" file contains functions for ascii\_to\_binary, int\_to\_binary, binary\_to\_ascii,

Add\_odd\_parity, remove\_odd\_parity which are required for encoding and decoding logic.

The "datalinklayer.c" file contains function for read\_file which is used to convert into frames and then add 2 SYN characters, add length of frame and encoded array of 32 characters into single frame.

int\_to\_binary function is used to convert integer to binary character

binary\_to\_ascii function is used to convert binary character to ascii character

add\_odd\_oarity function is used to add odd parity to the 8 bit code

remove\_odd\_parity function is used to remove odd parity from 8 bit char data

read\_file function is used to read file from file descriptor and convert them into frames of required length

The "encDec.h" header file contains function prototypes for the functions in "physicallayer.c" and "datalinklayer.c".

**Client:**

Client contain various methods like

crc\_division: Which is used to find crc remainder using given generator and append that remainder to the frame

calculate\_parity\_bit: Used to calculate the hamming code parity bit

hamming\_encode: Used to encode message bits by introducing hamming bits to the message data

Client creates a socket and reads input from intext.txt

Gives option to user to choose Error Mechanism No Error/CRC/Hamming

If No error chosen sends encoded data of read input data from intext.txt to server

If CRC chosen asks user to introduce error in which frame if chosen no continues else asks user to select which bit to introduce error then send encoded data to server

If Hamming chosen asks user to introduce error in which frame if chosen no continues else asks user to select which bit to introduce error then send encoded data to server

If response received from server decode the data and create result.txt and write changed data to result.txt

**Server:**

Server contain various methods like

crc\_division: Verify CRC\_divison

calculate\_parity\_bit: Used to calculate the hamming code parity bit

hamming\_decode: Used to decode hamming message bits by detecting error in bit level if error present in hamming message bits correct the bit error. And decode the message by removing hamming parity bits

Server creates socket and listens incoming client requests

If No error decode the data change the Vowels to upper case and append sum and then encode data and send response to client

If CRC chosen and no error decode the data change the Vowels to upper case and append sum and then encode data and send response to client

If CRC chosen and error present don’t send response to client

If Hamming chosen and no error present decode the data change the Vowels to upper case and append sum and then encode data and send response to client

If Hamming chosen and error present identify and correct bit error present in frame decode the data change the Vowels to upper case and append sum and then encode data and send response to client

**Test Documentation**

1.The project was tested on the provided intext.txt

**intext.txt**

Graphical user interface, text, application

Description automatically generated

**result.txt**

Graphical user interface, text, application

Description automatically generated

**2.** The project was tested on the data provided for project 2

**intext.txt**

Text

Description automatically generated

**result.txt**

Text

Description automatically generated

**User Documentation:**

1.Open the project folder in terminal and type the below lines to execute **server** first

Text

Description automatically generated

2. Open new terminal in project folder and type the below lines to execute **client** now

Text

Description automatically generated

**Selecting No error while transmitting from client:**

Client-side image:

Text

Description automatically generated

Server-side Image:

Text

Description automatically generated

**Selecting CRC error while transmitting from client:**

Client-side image:

Text

Description automatically generated

Server-side Image:

Text

Description automatically generated

**Selecting Hamming error while transmitting from client:**

Client-side Image:

Text

Description automatically generated

Server-side Image:

Text

Description automatically generated