**APPLICATION PROTOCOL**

**Architecture**

Client-server architecture is used for the network application. Server is given an IP address which does not change throughout the session. Client and server do not communicate directly to each other. They need to establish connection. Client requests services from the server and server responses to it.

* **Types of Messages**

**Request**: After a connection is established between the client and the server, client sends a request to the server. to translate the data of a file from one type to another and save it in the specified file.

**Response:** To answer to request of client, the server either sends client data or do the data translation job.

* **Syntax and Semantics of message type**
* **Request**

Format of the message

[file size] [data] [file format] [output filename size] [output filename]

[file size]: size of long

[data]: 1000 bytes of ASCII characters

[file format]: size of int

[output filename size]: size of int

[output filename]: (filename size) bytes of ASCII characters

Meaning of each field in the format

[file size]: size of the file

[data]: data of the file stored in buffer

[file format]: format number that specifies the different types of translation

[output filename size]: size of the output file name

[output filename]: name of the output file where the translated data are written

The value range of each field

**Size of primitive data types may differ in different platform**

[file size]: 0 to 4294967295

[data]: no specific range

[file format]: -32768 to 32767 or -2147483648 to 2147483647

[output filename size]: -32768 to 32767 or -2147483648 or 2147483647

[output filename]: no specific range

* **Response**

Format of the message

[file size] [ error message]

[file size]: size of long

[error message]: size of int

Meaning of each field in the format

[file size]: size of the file received by the client

[error message]: integer that tells whether the format of the units in the file was ok or not

The value range of each field

[file size]: 0 to 4294967295

[error message]: 0 or -1

* **Rules of sending messages**
* Client always sends the request to the server
* Server responds to the request of the server
* There is always one response for one request.
* Messages must follow the format and syntax mentioned above.
* If not follow, the program will throw an error.

**TEST CASES**

**USAGE OF CLIENT AND SERVER PROGRAM**

* **Usage of Client**

The following command invokes the client:

<client> <server IP> <server port> <file path> <to format> <to name>

Where

* <client> is the name of the client executable file name,
* <server IP> is the IP address of the server
* <server port> is the TCP of the server
* <file path> is the path of the file to be sent to the server (the file path indicates the location of the file in the system on which the server runs. It includes the file name, and possible the hierarchy of directories.)
* <to format> indicates how the server should translate the received units.

0 means no translation, 1 means to only translate type 0 units to type 1 with type 1 units unchanged, and 3 means to translate type 0 to 1 and type 1 to 0.

* <to name> is the name of the file the server should save the units to

If there are not enough or more than enough arguments, if the format is not in the specified range (0 to 3), then the client exits the program.