

CSCI-ECEN 4273-5273

Programming Assignment-1

UDP Socket Programming

**Submitted by:
Sridhar Pavithrapu**

I have implemented UDP protocol for transmitting information between a server and a client.

Here, I have created two folders ' clientFolder ' and ' serverFolder '.
The ' serverFolder ' consists of ' Makefile ', ' server.c ' and files for transfer.
Whereas the ' clientFolder ' consists of ' Makefile ' and ' client.c ' files.

Server:

In this assignment, I have written server which has below interfaces and their functionality:

server_get_file: This interface is used to send the file contents requested by client. First the server receives the file name needed from the client. Then server checks whether the file exists or not. If the file exists, then the server sends the file contents to the client.

server_put_file: This interface is used to store file contents received by client. First the client sends the size of the new file to be created. Then the server creates the new file and store the received contents in it.

server_delete_file: This interface is used to delete the file requested by client.

server_hash_value: This interface is used to print the hash value of the file requested by the client.

server_list_directory: This interface is used to list the files present in the directory and send its content to client upon request.

server_exit_server: This interface is used to exit the server upon request by client.

Client:

In this assignment, I have written client which has below interfaces and their functionality:

client_get_file: This interface is used to request file from server. First the client requests the file needed by sending the file name to the server. Then the client creates the new file and store the received contents in it.

client_put_file: This interface is used to send the file contents to server. First the client checks whether the file exists or not. If the file exists, then the client sends the file contents to the server.

client_delete_file: This interface is used to request server to delete the file.

client_hash_value: This interface is used to print the hash value of the file requested by the user.

client_list_directory: This interface is used to get the server directory contents.

client_exit_server: This interface is used to request server to exit.

Reliability:

In this assignment, the reliability is implemented using stop and wait protocol.

Here, first the sender sends the packet, then wait for a timeout to receive the acknowledgement. In this case I'm using packet number as acknowledgement. If I receive the required acknowledgement then I'll send the next packet, otherwise I'll send the same packet which is previously sent.

Encryption:

I have used 'XOR Encryption' for implementing encryption for the messages sent between server and client.

Executing Instructions:

Both the server and client consists of makefile. The server code and client code is compiled by running the 'make' command.

Client is executed by running the command:

First find the remote ip address. Then execute the below command:

./client 128.138.201.66 5000
[./client 'remote ip address' 'port number']

Server is executed by running the command:

./server 5000
[./server 'port number']

Once both the server and client is executed, user can see the menu for giving different operations as shown below:

```

/*****/
You can perform the following operations using this client:
1) Get File from the server.      [Eg: get 'file_name']
2) Put File in the server.       [Eg: put 'file_name']
3) Delete File in the server.    [Eg: delete 'file_name']
4) Get server contents by 'ls' command. [Eg: ls]
5) Get hash value of the file.    [Eg: md5sum 'file_name']
6) Exit the server.              [Eg: exit]
/*****/
Please enter the operation you want to perform:
```

For different operations, following commands should be given by the user:

- 1) For 'get' functionality:
 `get 'file_name'`
- 2) For 'put' functionality:
 `put 'file_name'`
- 3) For 'delete' functionality:
 `delete 'file_name'`
- 4) For server contents:
 `ls`
- 5) For hash value of the file:
 `md5um 'file_name'`
- 6) To exit the server:
 `exit`

After every operation performed, user will be prompted by the above menu options continuously until the exit command is given.