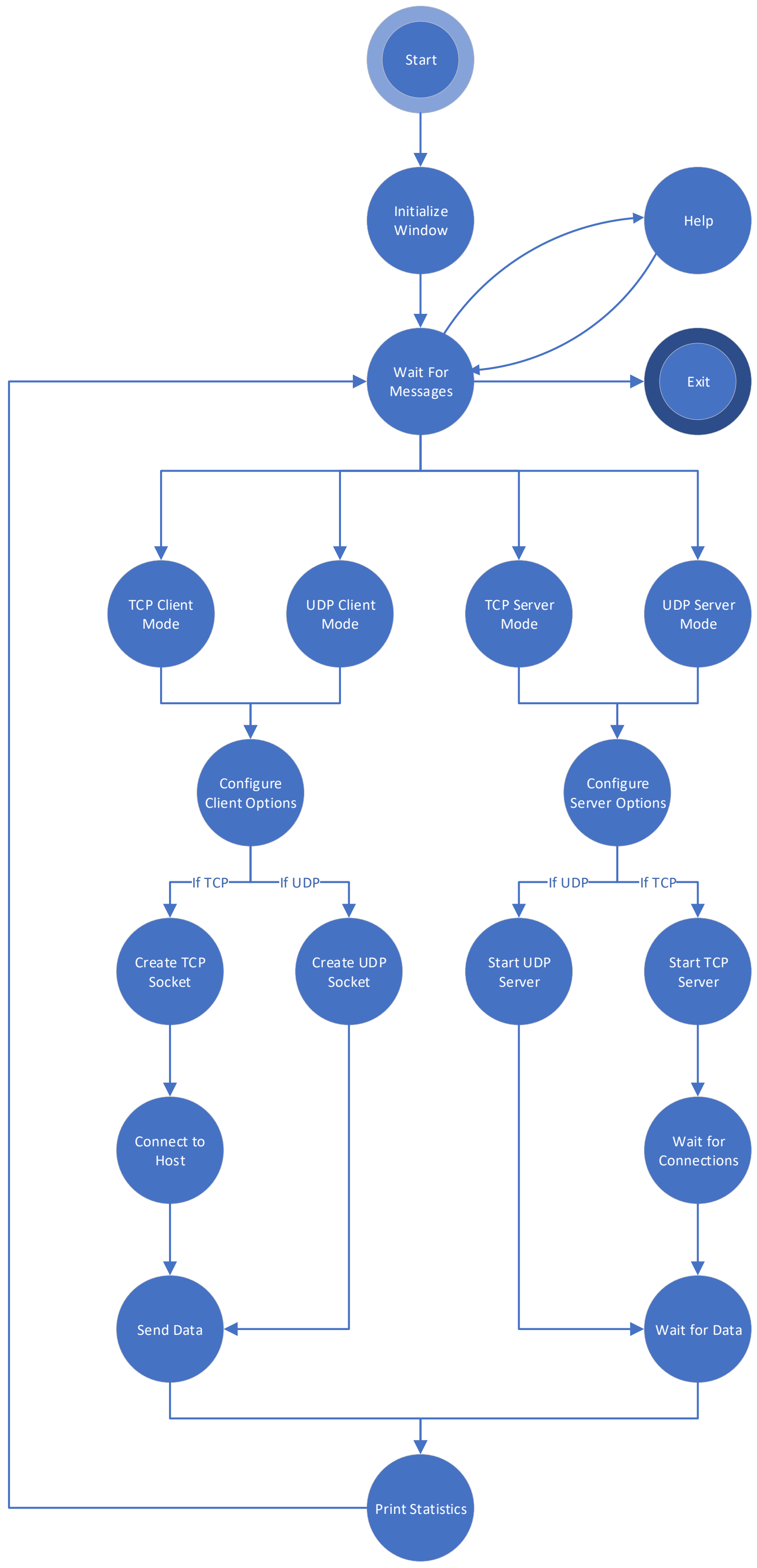
FILE TRANSFER/PROTOCOL ANALYSIS - DESIGN

COMP 4985

ASSIGNMENT 2 – FILE TRANSFER/PROTOCOL ANALYSIS

State Transition Diagram



Pseudocode

Initialize Window State

```
Initialize window;
Set window properties;
Initialize menu;

Show window;
Show menu;

If initializing window is successful
    Then enter Wait for Messages State;
```

Wait for Messages State

```
If user clicks “Help” menu item
    Then enter Help State;

If user clicks “Exit” menu item
    Then enter Exit State;

If user clicks “TCP Client”
    Then enter TCP Client State;

If user clicks “UDP Client”
    Then enter UDP Client State;

If user clicks “TCP Server”
    Then enter TCP Server State;

If user clicks “UDP Server”
    Then enter UDP Server State;
```

Help State

```
Initialize window;
Initialize help string;
Initialize “OK” button;

Show window in front of main window (parent window);
Print help string to window;
Show “OK” button;

If user clicks “OK” button
    Then enter Wait for Messages State;
```

Exit State

```
Deallocate variables;
Close all sockets;
Terminate program;
Close window;
```

TCP Client State

```
Set Mode to TCP Client;
Disable server operations;
If user clicks “Send Data” menu item
    Then enter Configure Client Options State;
```

UDP Client State

```
Set Mode to UDP Client;
Disable server operations;
If user clicks “Send Data” menu item
    Then enter Configure Client Options State;
```

TCP Server State

```
Set Mode to TCP Server;
Disable client operations;
If user clicks “Start Server” menu item
    Then enter Configure Server Options State;
```

UDP Server State

```
Set Mode to UDP Server;
Disable client operations;
If user clicks “Start Server” menu item
    Then enter Configure Server Options State;
```

Configure Client Options State

```
Initialize host string;
Initialize port string
Initialize packet size string;
Initialize number of packets string;

Get user inputs from each text box;
Save text to initialized strings;
Convert the saved text into proper types;

If the Protocol is TCP
    If user inputs all valid text fields
        Then enter Create TCP Socket State;

    If user inputs are invalid text fields
        Alert user of invalid text fields;
        Prompt user for input again;

If the Protocol is UDP
    If user inputs all valid text fields
        Then enter Create UDP Socket State

    If user inputs are invalid text fields
        Alter user of invalid text fields;
        Prompt user for input again;
```

Configure Server Options State

```
Initialize port string;

Get user input from text box;
Save text to the port string;
Convert port string to valid data type;

If the Protocol is TCP
    If the user inputs valid port
        Then enter Create TCP Socket State;

    If the user inputs invalid port
        Alert user of invalid port;
        Prompt user for input again;

If the Protocol is UDP
    If the user inputs valid port
        Then enter Create UDP Socket State;
    If the user inputs invalid port
        Alert user of invalid port;
        Prompt user for input again;
```

Create TCP Socket State

```
Create TCP socket;
Initialize host information from client options;

If the TCP socket was created successfully
    Then enter Connect to Host State;
```

Create UDP Socket State

```
Create UDP socket;
Initialize host information from client options;

If the TCP socket was created successfully
    Then enter Send Data State;
```

Start UDP Server State

```
Create UDP socket;
Initialize server address with the port from server options;

Bind the UDP socket to the address;
If the binding the UDP socket succeeds
    Then enter Wait for Data State;
```

Start TCP Server State

```
Create TCP socket;
Initialize server address with the information from server options;

Bind the TCP socket to the address;
Listen for connections coming from the TCP socket;
If listening for connections succeeds
    Then enter Wait for Connections State;
```

Connect to Host State

```
Connect TCP socket to the server side;

If the TCP socket has connected successfully
    Then enter Send Data State
```

Wait for Connections State

```
If a client chooses to connect to the server
    Accept the incoming connection;

    If accepting the client connection succeeds
        Then enter Wait for Data State;
```

Send Data State

```
Create Data to be sent;
Packetize data according to the client options;
Set Packet Size;
Set Number of Packets to send;
Create output string containing client info;

Send data through the socket;
If data is successfully sent to the server
    Then enter Print Statistics State;
```

Wait for Data State

```
Initialize a receive buffer;
Create output string containing server info;

If a client chooses to send data to the server
    Read the socket for incoming data;
    Save data to the receive buffer;

If the server received data
    Then enter Print Statistics State;
```

Print Statistics State

```
Get the output string created from the previous state;
Override application screen to blank;

Draw output string of text onto application main window

Enter Wait for Messages State;
```