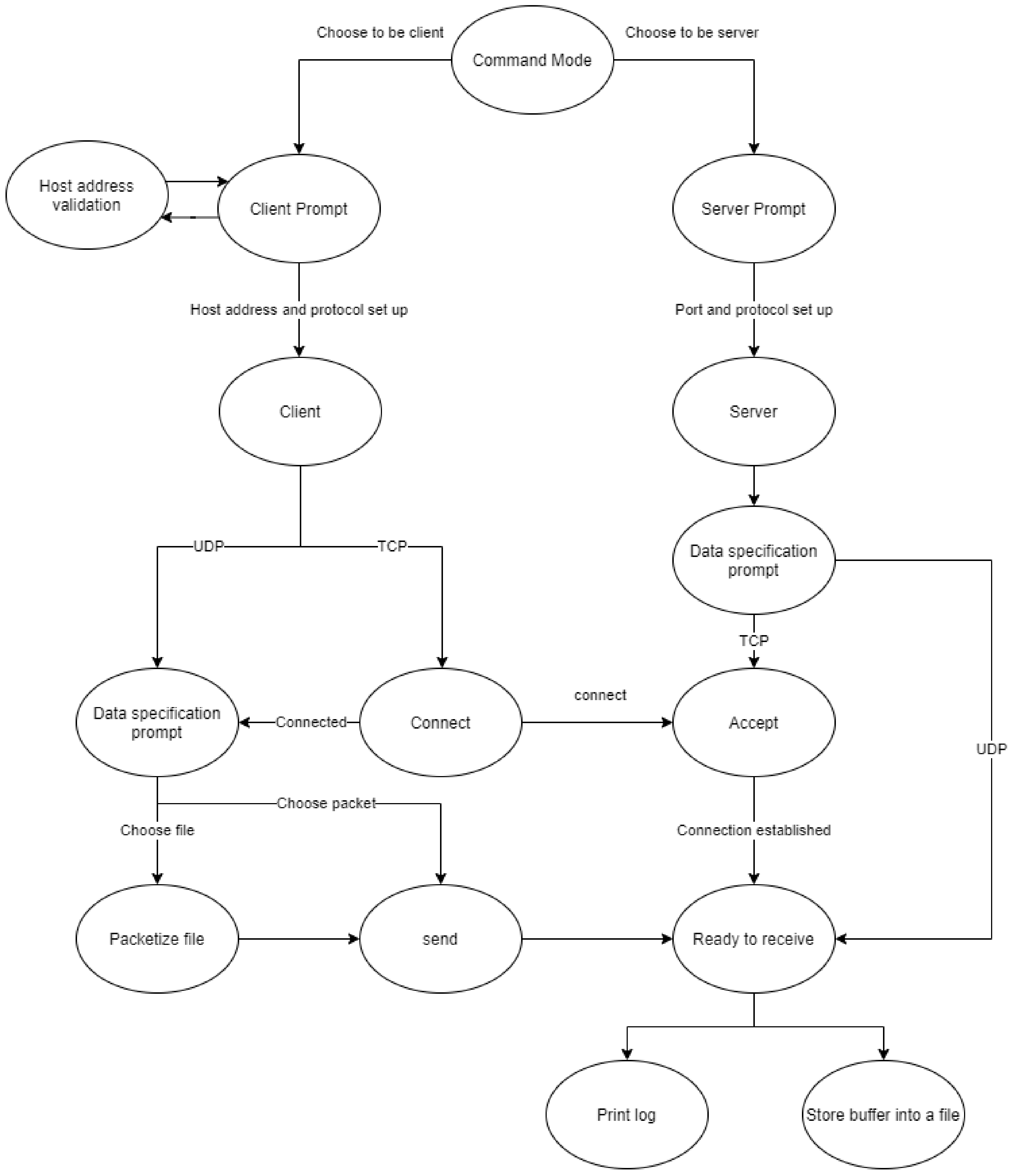
COMP 4985

Assignment 2

Design Document

Sam Lee
A01029480

State Diagram



Pseudo Code

Application Layer

Idle

```
WinMain {
    Register the window class with menu that has only client-server choice menu item
    activated.
    Create a window class with moderate size.
    Embed a text window in to the main window to display the data transmission log.
    Implement the window message handler.
}
WndProc {
    Handle window message depending on its type.
    Prepare instructions to open up the right dialog box according to the menu items.
}
```

Server - Print log

```
Print receiving information {
    Print the detail of communication with client
        - The time when the first packet is received
        - The time when the last packet is received
        - The total amount of received bytes
        - The total number of received packets
        - The size of the received packet
}
```

Client - Print log

```
Print sending information {
        Print the detail of the information of transmitting data
                - The time when the first packet is sent
                - The time when the last packet is sent
                - The total amount of sent bytes
                - The total number of sent packets
                - The size of the sent packet
}
```

Network layer

Server Info prompt

```
Protocol and Port configuration Dialog box for Server{
        Open a dialog box that prompts protocol and port number to use for networking.
        Store that information in application struct.
        Activate the server menu.
}
```

Client Info prompt

```
Protocol, port and IP configuration Dialog box for Client {
        Open a dialog box that prompts protocol and port number to use for networking
        Prompts host address and port number.
        Store that information in application struct.
        Activate the client menu.
        If the protocol is UDP, don't activate 'connect' menu
}
```

Client - Host address validation

```
IP validation{
        Start WSA
        Check if the host address is valid
        If the host address is valid
                Create client socket with the attribute depending on the protocol
                Store the host information in client struct
        If the host address is invalid
                Print error message on the client prompt dialog box
}
```

Client – Connect

```
TCP client connect {
        Connect to the stored host address
        If it fails return false
}
```

Server

```
Run TCP server{
        Start WSA
        Fill Addr info with server info
        Create a listen socket according to TCP specification
        Bind the socket with Addr struct
        Create a thread for the completion routine
        Listen for accepting a connection request
}
```

Server – Accept

```
TCP server accept {
        Accept a connection request
        Make accept socket
        Signal Accept event for actual reading with completion routine
}
```

Server – Ready to receive

```
Wait for accept event{
        Wait for the server thread to signal the accept event
}
Set completion routine{
        Fill the Socket info struct with the accept socket
        Trigger overlapped operation with that struct by passing it to completion routine
}
```

Data Link layer

Client – Data specification prompt

```
Data configuration{
        Open a dialog box that prompts data type, packet size and the number to send the
        packet.
        If the data type is a file, open a brose window to choose a text file to send.
}
```

Client – Packetize file

```
Packetize file{
    Packetize a text file according to the specification of data configuration
}
```

Server – Data specification prompt

```
Data configuration Dialog box {
    Open a dialog box that prompts expected data type, expected packet size and
    expected number of packets.
    If the data type is a file, open a brose window to choose a text file to store data.
}
```

Server – Depacketize

```
Store packets into a text file{
    Store received data into a text file
}
```

Physical layer

Client – Send

```
Send packets {
    Send packets or packetized file according to the protocol configured and data
    configuration.
    Keep updating the number of bytes sent
}
```

Server – Receive{

```
    Receive packets and handle it in completion routine
    Keep updating the number of bytes received
    Check the time when receiving the first packet and last packet
    Calculate the difference between the two times
    Keep reading the next packets
}
```