

Classes

Packet.cs

This defines the structure of a packet and provides methods for encoding and decoding them.

```
public Packet(byte[] raw)
```

- Creates a packet instance from the raw data from an incoming packet

```
public Packet(uint packetNumber, uint totalPackets, byte[] packetContent)
```

- Creates a packet from some content, the index, and the total packets in the group

PacketHandler.cs

This defines some methods for converting files and strings to a packet array

```
public static Packet[] FileToPackets(string filePath, uint packetSize)
```

- Reads in a file and creates a packet array from the data

```
public static void PacketsToFile(Packet[] packets, string filePath)
```

- Saves a packet array as a file with the given path

```
public static Packet[] ToPackets(byte[] data, uint packetSize)
```

- Converts a byte[] to a packet[]

```
public static byte[] FromPackets(Packet[] packets)
```

- Converts a packet[] to a byte[]

Tcp.cs

This defines methods for Sending a variety of types, beginning and ending receiving on a certain endpoint, and handling incoming packets.

```
public void Send(byte[] data, uint packetSize, EndPoint destination)
```

- Splits a byte[] into packets and sends it via Udp

```
public void SendFile(string filePath, uint packetSize, EndPoint destination)
```

- Reads in a file, converts it to packets, and sends it via Udp

```
public void SendMessage(string message, uint packetSize, EndPoint destination)
```

- Converts a string to a packets and sends it via Udp

```
public void BeginReceive(EndPoint remoteEndPoint)
```

- Starts listening on a particular ip and port

```
public void EndReceive()
```

- Stops listening

```
private void OnPacketReceived(Packet packet)
```

- Handles incoming packets

Udp.cs

This defines methods for send and async receive as well as defining a stateobject class to store data

```
public void SendPacket(Packet packet, EndPoint destination)
```

- Creates a temporary socket and sends one packet

```
public void BeginReceivePackets(EndPoint source)
```

- Begins an asynchronous callback loop which receives packets

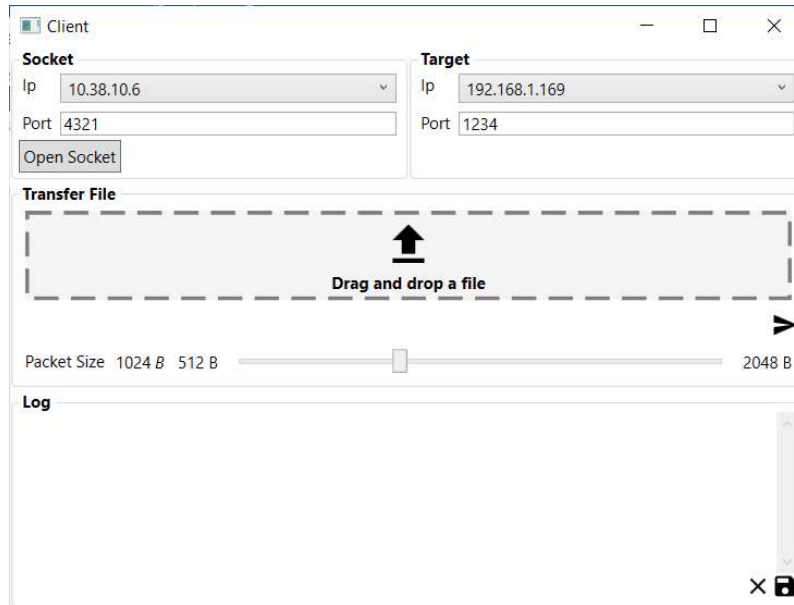
```
private void OnPacketReceived(IAsyncResult result)
```

- Recursive callback for receiving packets

Run Procedure

1. Launch both Client.exe and Server.exe and select target and socket ips for each.

(Note, the target of one must match socket of the other)



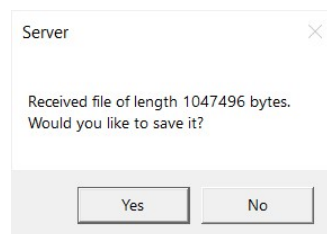
2. Click [Open Socket] to start the packet listener for each.
3. Drag and drop a bmp file into the Transfer File section.

(Note, only bitmaps can be dropped)

4. Select a packet size, default of 1KiB
5. Click the send button in the bottom right of the transfer section to send the file.

(Note, this button will be disabled until a file is dropped)

6. Look for the popup from the other side.



7. If you like, you can save the incoming bmp file. If not, simply click no.
8. In the log section, there are two Icons. Save and Clear.
 - a. Save will open a save file prompt in which the user can enter a filename and the log file will be saved.
 - b. Clear will clear the log file.
9. When finished, close both windows.

