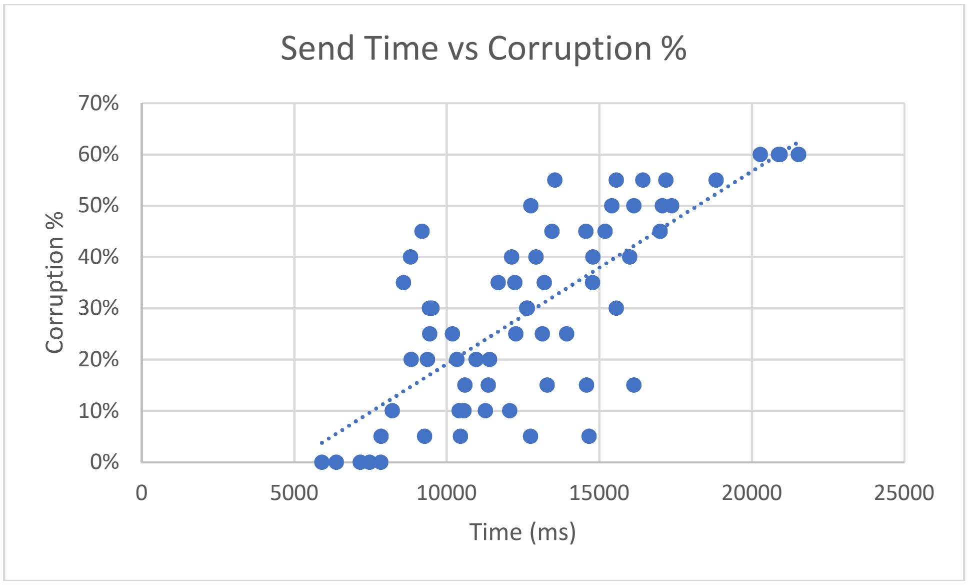
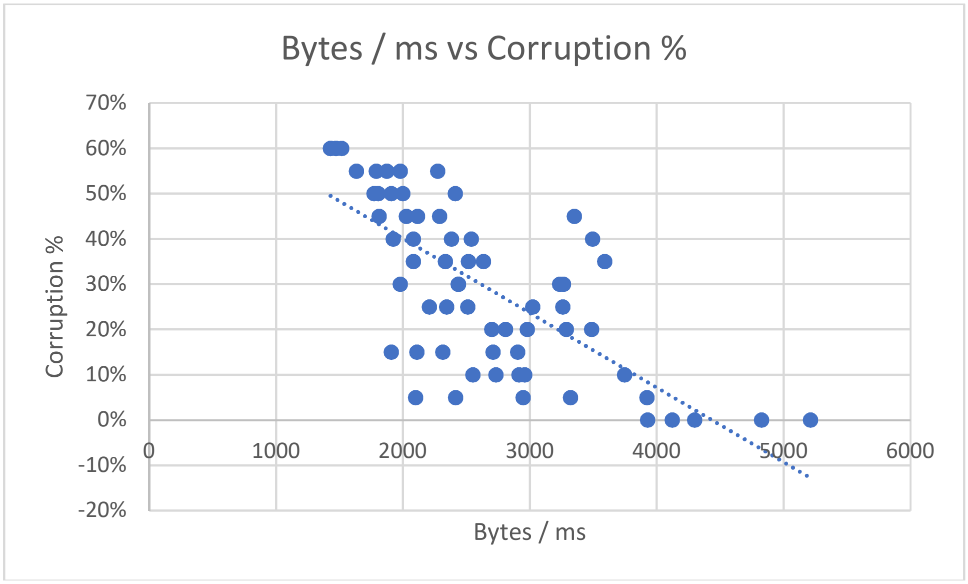
Objective

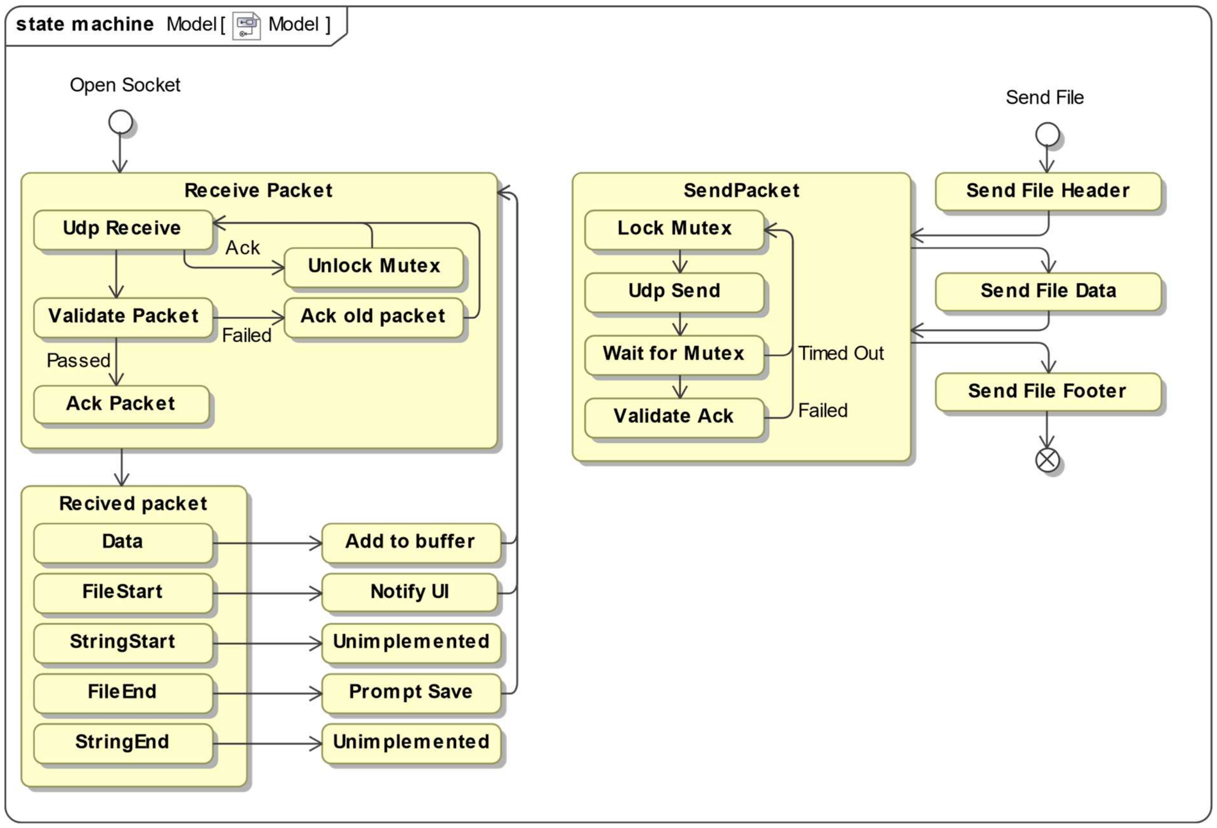
The objective of this phase is to improve phase 2 and implement RDT 2.2. Firstly, phase 3 is to add checksum as a header to the sending data, which checks the packets and make sure the data is not corrupted. Another objective is to add the sequence number of the packets and add it as a header to the sending data. The sender needs to resend a packet if it receives a ACK with old sequence number or when this time runs out.

File delivery times with corruption





Rdt State Machine



<u>Class and function Descriptions</u>

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(byte sequenceNumber, byte[] packetContent, Packtype packType = PackType.Data)

- Creates a packet from data, packet number, and total packets

public byte[] Raw;

- Holds the raw data for the packet

public byte SequenceNumber

- Returns the packet's sequence number

public PacketType TypeByte

- Returns the packet's type

public byte GetCheckSum()

- Calculates the packet's checksum

public bool IsValidChecksum()

- Returns true if the current checksum is valid

public byte[] GetNonHeaderData()

- Returns the raw packet data without header

Public enum PacketType

- Possible packet types

PacketHandler.cs

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

public static Packet[] FileToPackets(string filePath, int packetSize, byte currentSequenceNumber)

- 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[] StringToPackets(string str, int packetSize, ref byte currentSequenceNumber)

- Converts a string to a packet array

public static string PacketsToString(Packet[] packets)

- Converts a packet array to a string

public static Packet[] ToPackets(byte[] data, int packetSize, byte currentSequenceNumber)

- Converts a byte array into packets array

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

- Converts a packets array to a byte array

Rdt.cs

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

private void SendPacket(Packet packet, EndPoint remoteEndPoint)

- Sends a single packet with ack checking

private void Send(byte[] data, EndPoint remoteEndPoint, int packetSize = 1024)

- Breaks a byte array into packets of size <packetSize> and sends them

public void SendFile(string filePath, EndPoint remoteEndPoint, int packetSize = 1024)

- Reads in a file, breaks it into packets, and sequentially sends them to the given endpoint

public vois SetCorruptionOption(bool dataPacketCorruptionEnable, double dataPacketCorruptionChance, bool ackPacketCorruptionEnabled, double ackPacketCorruptionChance)

- Sets the corruption options of Rdt

private bool percentToBoolRand(double percentChance)

- Returns true or false depending on percentChance

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)

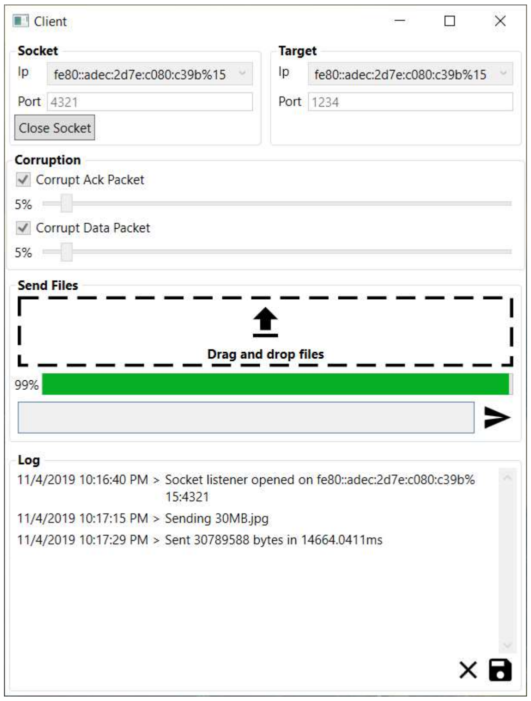
- Begins an asynchronous callback loop which receives packets

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)



Client

Socket
Ip fe80::adec:2d7e:c080:c39b%15
Port 4321
Close Socket

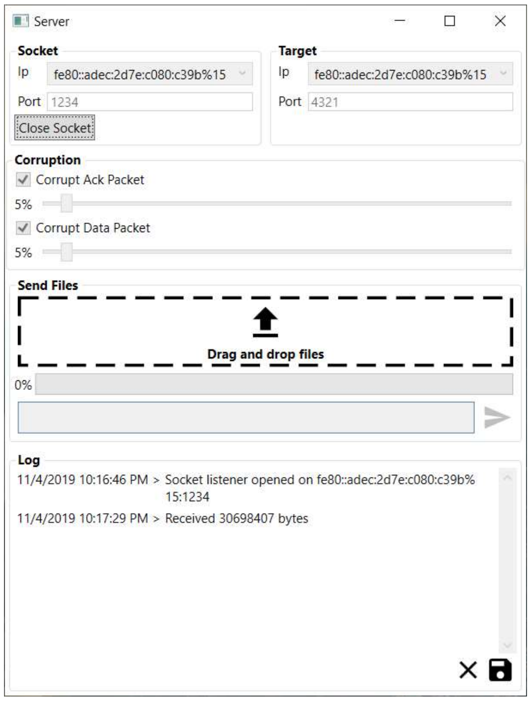
Target
Ip fe80::adec:2d7e:c080:c39b%15
Port 1234

Corruption
✓ Corrupt Ack Packet
5%
✓ Corrupt Data Packet
5%

Send Files
Drag and drop files
99%

Log

```
11/4/2019 10:16:40 PM > Socket listener opened on fe80::adec:2d7e:c080:c39b%
                        15:4321
11/4/2019 10:17:15 PM > Sending 30MB.jpg
11/4/2019 10:17:29 PM > Sent 30789588 bytes in 14664.0411ms
```



Server

Socket
Ip fe80::adec:2d7e:c080:c39b%15
Port 1234
Close Socket

Target
Ip fe80::adec:2d7e:c080:c39b%15
Port 4321

Corruption
✓ Corrupt Ack Packet
5%
✓ Corrupt Data Packet
5%

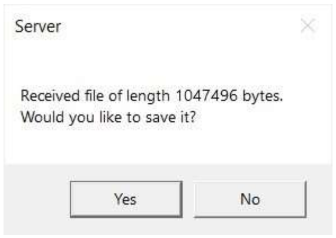
Send Files
Drag and drop files
0%

Log

```
11/4/2019 10:16:46 PM > Socket listener opened on fe80::adec:2d7e:c080:c39b%
                        15:1234
11/4/2019 10:17:29 PM > Received 30698407 bytes
```

2. Select your corruption settings.
3. Click [Open Socket] to start the packet listener for each.
4. Drag and drop the provided jpg file into the Transfer File section.
5. Click the send button in the bottom right of the transfer section to send the file.

6. Look for the popup from the other side.



Server

Received file of length 1047496 bytes.
Would you like to save it?

Yes No

7. If you like, you can save the incoming jpg 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.