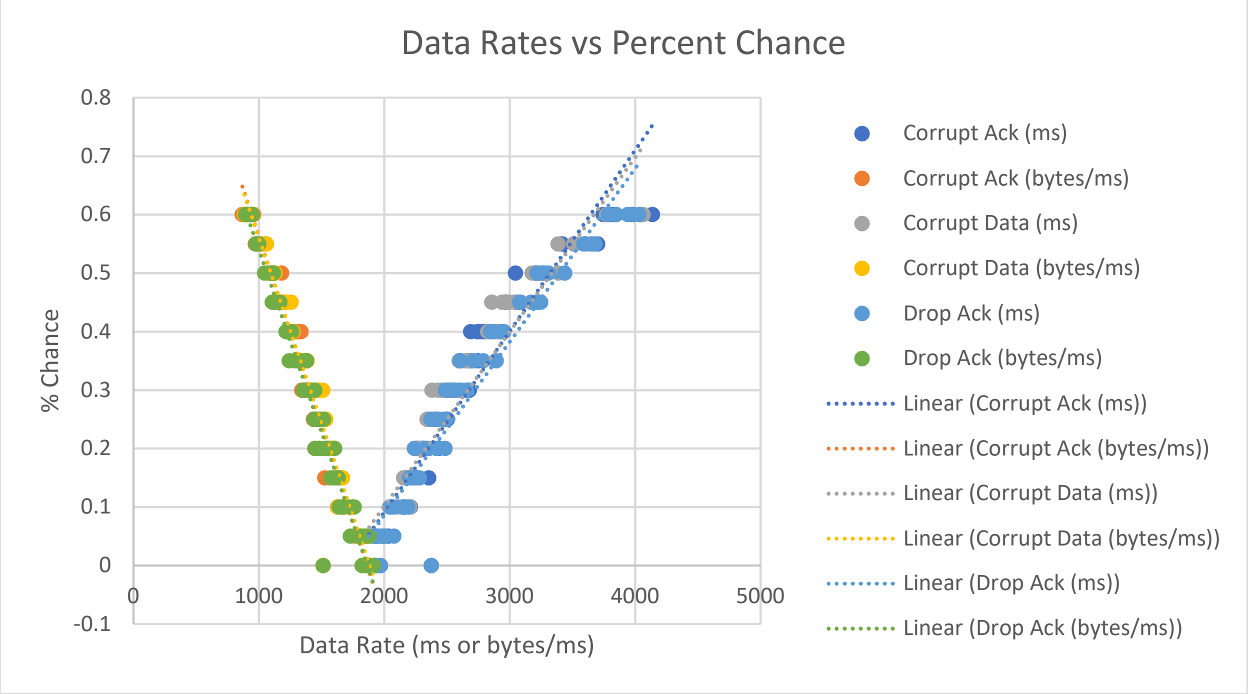
Objective

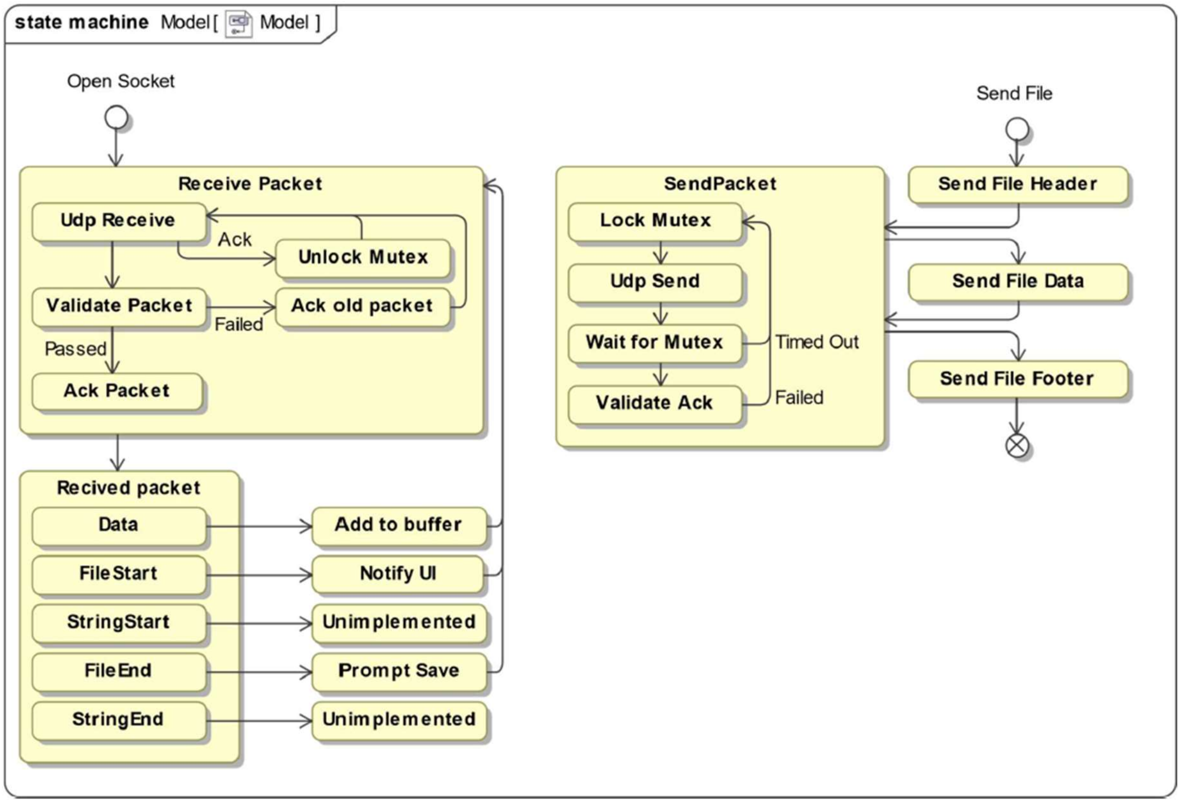
The objective of this phase is to improve phase 3 and implement RDT 3.0. RDT 3.0 is to enhance the sender to handle the data packet lost and ACK packet lost. In phase 4, the timer will be the main control to handle the data packet loss, and ACK packet loss. The data will be resent after every clock cycle if it is not reset by the ACK with the right sequence number from the receiver. The timer will be restarted every time the packet is sent and becomes the main control over the packet sending.

File Delivery times with corruption



Note: We had some data for Data Drop but it became prohibitively slow to run the tests.

RDT State Machine



Class and Function Descriptions

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

public Packet(byte[] raw)

- Creates a packet from raw data

public Packet(byte sequenceNumber, byte[] packetContent, PacketType packetType = PacketType.Data)

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

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 (not including the header)

public Packet AsCorrupt()

- Returns a corrupted copy of the packet

public enum PacketType

- Possible packet types

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

public void SetCorruptionOptions(NetDesign_UC_Lib.PacketErrorInfo packetErrorOptions)

- Sets the corruption options of Rdt

public void SendFilesAsync(string[] filePaths, EndPoint destination, int packetSize = 1024)

- Send files asynchronously

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

- Send file to the file path, the destination with the packet sizes of 1024

private void SendDataPacket(Packet packet, EndPoint destination)

- Send created packet to the destination]

public void BeginReceive(ushort port)

- Begin the receiver of the UDP

private void Udp_PacketReceived(Packet packet, EndPoint senderEndPoint, EndPoint receiverEndPoint)

- It will handle incoming ACK, Corrupt data packet at receiver, drop data packet at receiver, check incoming data, create and send ACK

private void Rdt_PacketReceived(Packet packet, EndPoint senderEndPoint, EndPoint receiverEndPoint)

- Check the type of packet received and handle the packet

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 BeginReceive(ushort port)

- Resets the socket, Binds the socket to the port, create a new empty endpoint, waits for a packet asynchronously

private void PacketReceivedCallback(IAsyncResult result)

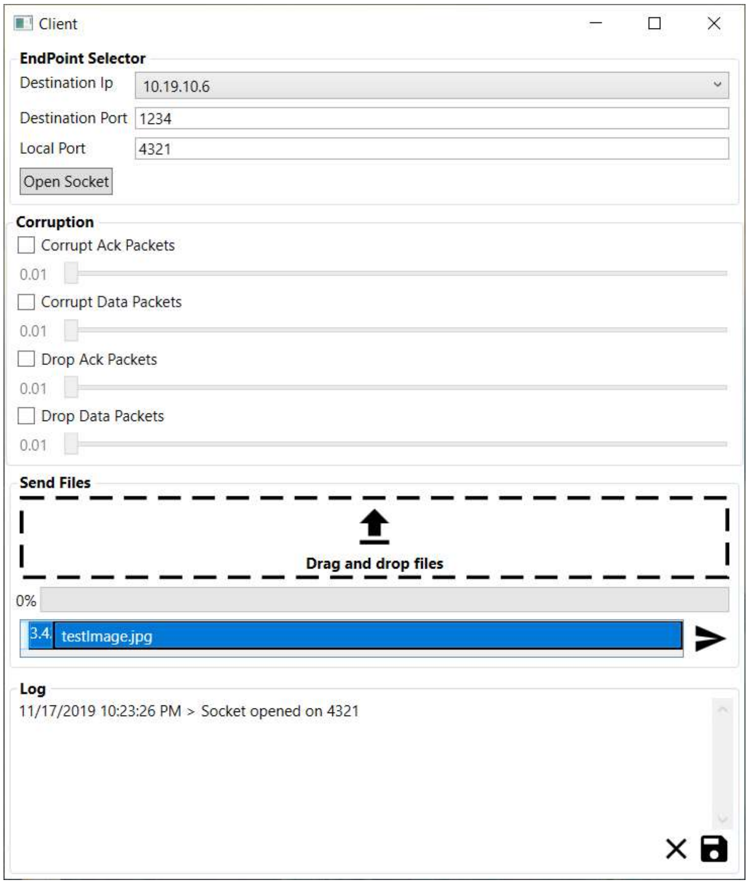
- Gets the passed socket, sender's endpoint, number of transferes bytes, packet from the buffer and also create a new empty endpoint and waits for a packet asynchronously, and deliver the packet.

public void SendPacket(Packet packet, EndPoint destination)

- Sends the packet

Run Procedure

1. Launch both Client.exe and Server.exe and select ips for each. (Note, ipv6 is not currently working)



Destination Ip 10.19.10.6

Destination Port 1234

Local Port 4321

☐ Corrupt Ack Packets 0.01

☐ Corrupt Data Packets 0.01

☐ Drop Ack Packets 0.01

☐ Drop Data Packets 0.01

```
11/17/2019 10:23:26 PM > Socket opened on 4321
```

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.

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.