

UDP File Transfer with RDT 2.2 - Design Document

UDP File Transfer with RDT 2.2

Author: Ayush Pandey

Date: 02-21-2025

1. Introduction

This document describes the design and implementation of a file transfer system using UDP sockets and the RDT 2.2 protocol. The system consists of a client and a server that communicate over a network to transfer a JPEG image file.

2. Project Structure

The project is divided into two main sections:

- **Section 1(a):** RDT 2.2 sender

stateDiagram-v2

state "Wait for call 0 from above" as WaitCall0

state "Wait for ACK 0" as WaitACK0

state "Wait for ACK 1" as WaitACK1

state "Wait for call 1 from above" as WaitCall1

WaitCall0 → WaitACK0 : rdt_send(data)\nsndpkt make_pkt (0, data, checksum)\nudt_send(sndpkt)

WaitACK0 → WaitACK0 : rdt_rev (rcvpkt) &&\n(corrupt (revpkt) || \nisACK (revpkt.1))\nudt_send (sndpkt)

WaitACK0 → WaitCall1 : rdt_rev (rcvpkt) &&\nnot.corrupt (revpkt) &&\nisACK (rcvpkt.1)

WaitCall1 → WaitACK1 : rdt_send(data)\nsndpkt make_pkt (1, data, checksum)\nudt_send(sndpkt)

```
WaitACK1 → WaitACK1 : rdt_rev(revpkt) &&\n(corrupt(revpkt)) || \nISACK (revpkt,0))\nudt_send (sndpkt)
```

```
WaitACK1 → WaitCall0 : rdt_rev(revpkt) &&\nnot.corrupt (revpkt) &&\nisACK (revpkt, 0)
```

- **Section 1(b):** RDT 2.2 receiver

```
stateDiagram-v2
```

```
state "Wait for 0 from below" as Wait0FromBelow
```

```
state "Wait for 1 from below" as Wait1FromBelow
```

```
Wait0FromBelow → Wait0FromBelow : rdt_rev (revpkt) &&\n(corrupt (revpkt)) || \nhas_neq1 (revpkt))\nif (oncethru==1)\nudt_send(sndpkt)
```

```
Wait0FromBelow → Wait1FromBelow : rdt_rev (revpkt) &&\nnotcorrupt (revpkt) &&\nhas_seq0(rcvpkt)\nextract (revpkt.data)\ndeliver_data (data)\nsndpkt make_pkt (ACK, 0, checksum)\nudt_send(sndpkt.)\noncethru=1
```

```
Wait1FromBelow → Wait1FromBelow : rdt_rev (revpkt) &&\n(corrupt (revpkt)) || \nhas_seq0(revpkt))\nudt_send(sndpkt)
```

```
Wait1FromBelow → Wait0FromBelow : rdt_rev(revpkt) &&\nnotcorrupt (revpkt) &&\nhas_seq1(rcvpkt)\nextract (revpkt, data)\ndeliver_data (data)\nsndpkt make_pkt (ACK, 1, checksum)\nudt_send (sndpkt)\noncethru=0
```

3. Section 1(c): Use binary variables for representing bits (instead of strings)

In this phase, we will need to implement ALL the following data transfer scenarios:

Client Implementation

```
#client_phase_2.py
```

```
import socket
import struct
import random
```

```

from file_transfer import send_file
# Server details
SERVER_ADDRESS = ('localhost', 12351) # Use the same port number as the ser

# Packet size
PACKET_SIZE = 1024

def calculate_checksum(data):
    # Calculate the checksum using XOR
    checksum = 0
    for byte in data:
        checksum ^= byte
    return checksum

def introduce_bit_error(data):
    # Introduce a bit error in the data
    if len(data) > 0:
        byte_index = random.randint(0, len(data) - 1)
        bit_index = random.randint(0, 7)
        data = bytearray(data)
        data[byte_index] ^= 1 << bit_index
    return bytes(data)

def send_file(file_path, update_fsm_state, option=1, error_rate=0.0):
    # Create a UDP socket
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    client_socket.settimeout(10) # Set a timeout for the socket

    # Read the file data
    with open(file_path, "rb") as f:
        file_data = f.read()

    # Calculate the number of packets
    num_packets = (len(file_data) + PACKET_SIZE - 1) // PACKET_SIZE

    # Send the number of packets to the server

```

```

client_socket.sendto(struct.pack("!I", num_packets), SERVER_ADDRESS)
update_fsm_state("Sent number of packets")
print("Sent number of packets")

# Send the file data in packets
seq_num = 0
for i in range(num_packets):
    start = i * PACKET_SIZE
    end = start + PACKET_SIZE
    packet_data = file_data[start:end]
    checksum = calculate_checksum(packet_data)
    packet = struct.pack("!II" + str(len(packet_data)) + "s", seq_num, checksum, packet_data)
    client_socket.sendto(packet, SERVER_ADDRESS)
    update_fsm_state(f"Sent packet {i} with sequence number {seq_num}")
    print(f"Sent packet {i} with sequence number {seq_num}")

# Wait for ACK
try:
    ack, _ = client_socket.recvfrom(2048)
    if len(ack) == 8: # Ensure the ACK packet is 8 bytes long
        ack_seq_num, ack_checksum = struct.unpack("!II", ack)

        # Option 2: Introduce bit error in ACK packet for testing
        if option == 2 and random.random() < error_rate:
            ack_seq_num = introduce_bit_error(struct.pack("!I", ack_seq_num))
            ack_seq_num = struct.unpack("!I", ack_seq_num)[0]

        if ack_seq_num != seq_num:
            print(f"Error: Incorrect ACK sequence number. Expected {seq_num},
            continue # Handle retransmission or other error recovery mechanism
        else:
            print("Error: Received malformed ACK packet")
            continue # Handle retransmission or other error recovery mechanisms
except socket.timeout:
    print("Timeout waiting for ACK")
    continue # Handle retransmission or other error recovery mechanisms

```

```

except ConnectionResetError as e:
    print(f"ConnectionResetError: {e}")
    break

seq_num = 1 - seq_num # Toggle sequence number

# Close the socket
client_socket.close()
update_fsm_state("File sent successfully")
print("File sent successfully")

# Example usage
# filename = "phase_2.jpg" # image file
# send_file(filename, update_fsm_state, option=1, error_rate=0.0) # Option 1: No

```

Server Implementation

```

#server_phase_2.py

import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk
import threading
import time
from file_transfer import receive_file
# Import the send_file function from client_phase_2.py
import client_phase_2
# Import the receive_file function from server_phase_2.py
import server_phase_2

def start_receiver():
    # Start the receive_file function in a separate thread
    threading.Thread(target=receive_file_with_status, daemon=True).start()

def receive_file_with_status():

```

```

try:
    server_phase_2.receive_file(update_fsm_state)
    status_label.config(text="File received successfully!")
    display_image("received_image.jpg")
except Exception as e:
    status_label.config(text=f"Error receiving file: {e}")
    print(f"Error receiving file: {e}")

def send_file_with_status():
    file_path = filedialog.askopenfilename()
    if file_path:
        try:
            client_phase_2.send_file(file_path, update_fsm_state)
            status_label.config(text="File sent successfully!")
            display_image(file_path)
        except Exception as e:
            status_label.config(text=f"Error sending file: {e}")
            print(f"Error sending file: {e}")

def display_image(file_path):
    try:
        img = Image.open(file_path)
        img = img.resize((300, 300), Image.LANCZOS) # Use Image.LANCZOS inst
        img = ImageTk.PhotoImage(img)
        image_label.config(image=img)
        image_label.image = img
    except Exception as e:
        messagebox.showerror("Error", f"Unable to display image: {e}")
        print(f"Unable to display image: {e}")

def update_fsm_state(state):
    fsm_label.config(text=f"FSM State: {state}")
    print(f"FSM State: {state}")

# Create the main window
window = tk.Tk()

```

```

window.title("RDT 2.2 File Transfer")
window.geometry("600×600")

# Create a button to select and send the file
send_button = tk.Button(window, text="Send File", command=send_file_with_stat
send_button.pack(pady=20)

# Create a button to start receiving the file
receive_button = tk.Button(window, text="Receive File", command=start_receive
receive_button.pack(pady=20)

# Create a label to display the status
status_label = tk.Label(window, text="", font=("Arial", 12))
status_label.pack(pady=10)

# Create a label to display the image
image_label = tk.Label(window)
image_label.pack(pady=10)

# Create a label to display the FSM state
fsm_label = tk.Label(window, text="FSM State: Idle", font=("Arial", 12))
fsm_label.pack(pady=10)

# Run the main loop
window.mainloop()

```

File Transfer Implementation

```

#file_transfer.py
import socket
import struct
import random
import time

def calculate_checksum(data):

```

```

checksum = 0
for byte in data:
    checksum += byte
    checksum &= 0xFF # Keep it 8-bit
return ~checksum & 0xFF

def make_rdt_packet(seq_num, data):
    checksum = calculate_checksum(data)
    header = struct.pack('! B', seq_num, checksum)
    return header + data

def parse_rdt_packet(packet):
    header = packet[:5]
    data = packet[5:]
    seq_num, checksum = struct.unpack('! B', header)
    return seq_num, checksum, data

def is_corrupt(packet):
    seq_num, checksum, data = parse_rdt_packet(packet)
    return calculate_checksum(data) != checksum

def make_ack_packet(seq_num):
    checksum = calculate_checksum(struct.pack('! B', seq_num))
    return struct.pack('! B', seq_num, checksum)

def parse_ack_packet(packet):
    return struct.unpack('! B', packet)

def make_packet(file_path, packet_size=1024):
    packets = []
    with open(file_path, 'rb') as f:
        while True:
            chunk = f.read(packet_size)
            if not chunk:
                break
            packets.append(chunk)

```



```

    return packets

def create_udp_socket():
    return socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

def send_packet(sock, packet, address):
    sock.sendto(packet, address)

def receive_packet(sock, buffer_size=1030): # Increased buffer size to 1030
    (1024 + 6 for header)
    return sock.recvfrom(buffer_size)

# Define the address and listen_address variables
address = ('localhost', 12345)
listen_address = ('localhost', 12345)
save_path = 'C:/Users/Ayush_Pandey/Dev/RDT/received_image.jpg'

def send_file(file_path, update_fsm_state):
    packets = make_packet(file_path)
    sock = create_udp_socket()
    seq_num = 0

    for packet in packets:
        rdt_packet = make_rdt_packet(seq_num, packet)
        try:
            send_packet(sock, rdt_packet, address)
            print(f"Sent packet {seq_num}")
            while True:
                ack_packet, _ = receive_packet(sock)
                ack_seq_num, ack_checksum = parse_ack_packet(ack_packet)
                if ack_seq_num == seq_num and not is_corrupt(ack_packet):
                    print(f"Received ACK for packet {seq_num}")
                    break
            except Exception as e:
                print(f"Error sending packet {seq_num}: {e}")
                update_fsm_state(f"Error sending packet {seq_num}: {e}")

```

```

        return
    seq_num += 1
    update_fsm_state(f"Sent packet {seq_num}")

# Send an empty packet to indicate the end of the file transfer
end_packet = make_rdt_packet(seq_num, b'')
send_packet(sock, end_packet, address)
print("Sent end of file packet")
update_fsm_state("Sent end of file packet")

def receive_file(update_fsm_state):
    sock = create_udp_socket()
    sock.bind(listen_address)
    received_packets = {}
    expected_seq_num = 0

    while True:
        try:
            rdt_packet, sender_address = receive_packet(sock)
            if not is_corrupt(rdt_packet):
                seq_num, _, data = parse_rdt_packet(rdt_packet)
                if seq_num == expected_seq_num:
                    if data == b'': # End of file packet
                        print("Received end of file packet")
                        update_fsm_state("Received end of file packet")
                        break
                    received_packets[seq_num] = data
                    ack_packet = make_ack_packet(seq_num)
                    send_packet(sock, ack_packet, sender_address)
                    print(f"Received packet {seq_num}")
                    expected_seq_num += 1
                    update_fsm_state(f"Received packet {seq_num}")
                else:
                    ack_packet = make_ack_packet(expected_seq_num - 1)
                    send_packet(sock, ack_packet, sender_address)
            else:

```

```

        ack_packet = make_ack_packet(expected_seq_num - 1)
        send_packet(sock, ack_packet, sender_address)
    except Exception as e:
        print(f"Error receiving packet: {e}")
        update_fsm_state(f"Error receiving packet: {e}")
        return

    with open(save_path, 'wb') as f:
        for i in range(expected_seq_num):
            f.write(received_packets[i])
    print("File received successfully")
    update_fsm_state("File received successfully")

```

UI Implementation

```

#ui_phase_2
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk
import threading
import time
import matplotlib.pyplot as plt

# Import the send_file and receive_file functions from file_transfer.py
from file_transfer import send_file, receive_file

def start_receiver():
    # Start the receive_file function in a separate thread
    threading.Thread(target=receive_file_with_status, daemon=True).start()

def receive_file_with_status():
    try:
        update_fsm_state("Receiving file...")
        start_time = time.time()

```

```

        receive_file(update_fsm_state)
        end_time = time.time()
        transfer_time = end_time - start_time
        status_label.config(text=f"File received successfully in {transfer_time:.2f}
seconds!")
        display_image("received_image.jpg")
        window.after(0, plot_performance, transfer_time, "Receive")
    except Exception as e:
        status_label.config(text=f"Error receiving file: {e}")
        print(f"Error receiving file: {e}")

def send_file_with_status():
    file_path = filedialog.askopenfilename()
    if file_path:
        threading.Thread(target=send_file_thread, args=(file_path,), daemon=True).start()

def send_file_thread(file_path, retry_count=3):
    try:
        update_fsm_state("Sending file...")
        start_time = time.time()
        send_file(file_path, update_fsm_state)
        end_time = time.time()
        transfer_time = end_time - start_time
        status_label.config(text=f"File sent successfully in {transfer_time:.2f} seconds!")
        display_image(file_path)
        window.after(0, plot_performance, transfer_time, "Send")
    except Exception as e:
        status_label.config(text=f"Error sending file: {e}")
        print(f"Error sending file: {e}")
        # Additional logging for debugging
        with open("error_log.txt", "a") as log_file:
            log_file.write(f"Error sending file: {e}\n")
        # Attempt to reconnect or handle the error

```

```

        handle_connection_error(e, file_path, retry_count)

def handle_connection_error(error, file_path, retry_count):
    if isinstance(error, ConnectionResetError) and retry_count > 0:
        print("Connection was reset. Attempting to reconnect...")
        # Implement reconnection logic here
        # Retry sending the file after a short delay
        time.sleep(5)
        print(f"Retrying... {retry_count} attempts left.")
        send_file_thread(file_path, retry_count - 1)
    else:
        print(f"Unhandled error or no retries left: {error}")
        # Additional logging for debugging
        with open("error_log.txt", "a") as log_file:
            log_file.write(f"Unhandled error or no retries left: {error}\n")

def display_image(file_path):
    try:
        img = Image.open(file_path)
        img = img.resize((300, 300), Image.LANCZOS) # Use Image.LANCZOS in
        #stead of Image.ANTIALIAS
        img = ImageTk.PhotoImage(img)
        image_label.config(image=img)
        image_label.image = img
    except Exception as e:
        messagebox.showerror("Error", f"Unable to display image: {e}")
        print(f"Unable to display image: {e}")

def update_fsm_state(state):
    fsm_label.config(text=f"FSM State: {state}")
    print(f"FSM State: {state}")

def plot_performance(transfer_time, operation):
    plt.figure()
    plt.bar(operation, transfer_time)

```

```

plt.xlabel('Operation')
plt.ylabel('Transfer Time (s)')
plt.title('File Transfer Performance')
plt.show()

# Create the main window
window = tk.Tk()
window.title("RDT 2.2 File Transfer")
window.geometry("600×600")

# Create a button to select and send the file
send_button = tk.Button(window, text="Send File", command=send_file_with_s
tatus, font=("Arial", 14))
send_button.pack(pady=20)

# Create a button to start receiving the file
receive_button = tk.Button(window, text="Receive File", command=start_recei
ver, font=("Arial", 14))
receive_button.pack(pady=20)

# Create a label to display the status
status_label = tk.Label(window, text="", font=("Arial", 12))
status_label.pack(pady=10)

# Create a label to display the image
image_label = tk.Label(window)
image_label.pack(pady=10)

# Create a label to display the FSM state
fsm_label = tk.Label(window, text="FSM State: Idle", font=("Arial", 12))
fsm_label.pack(pady=10)

# Run the main loop
window.mainloop()

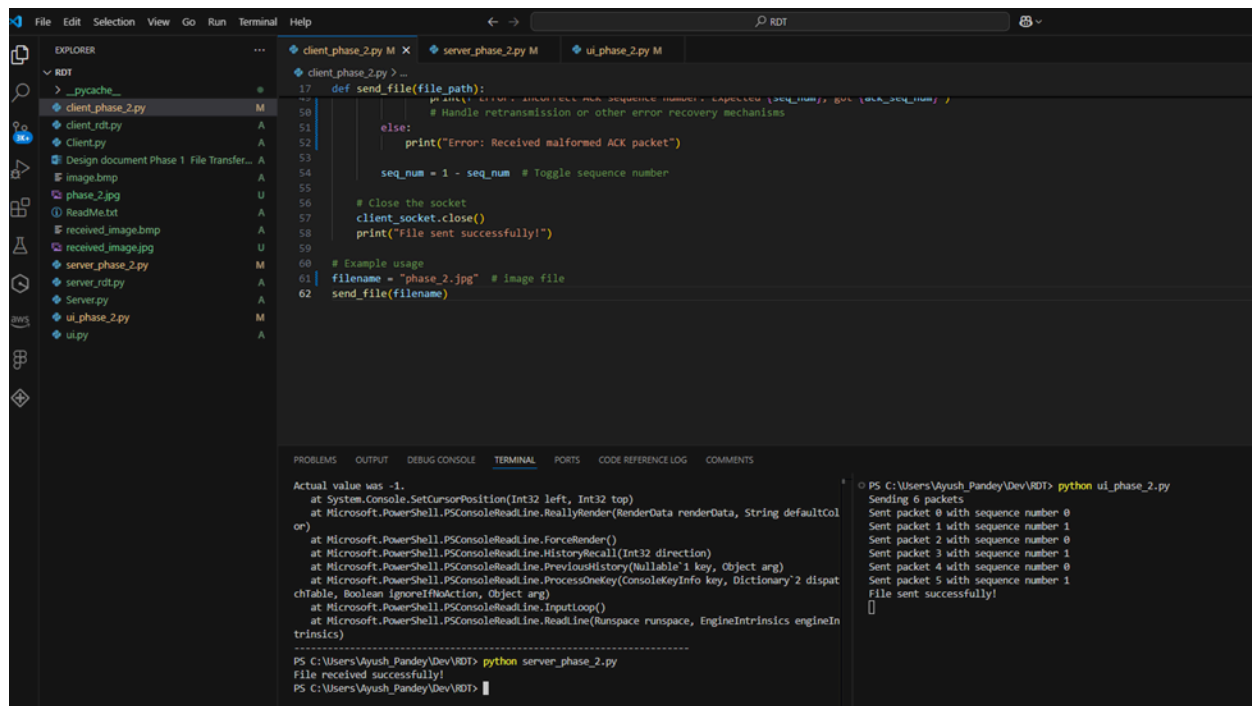
```

4. Sample Scenarios

Section 1(a): Basic implementation

Terminal 1 (Server):

```
$ python server_phase_2.py  
File received successfully!
```



Terminal 2 (Client):

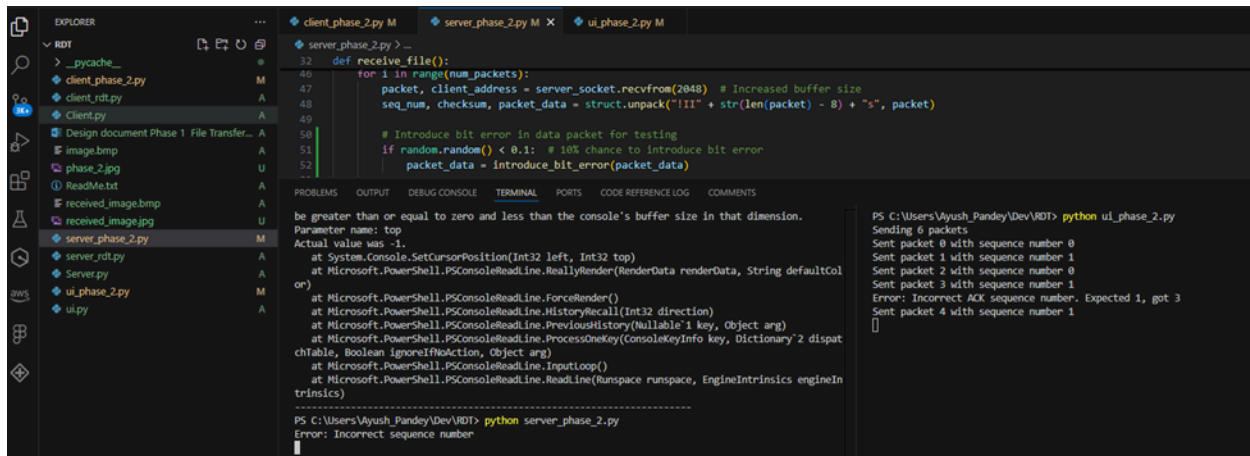
```
$ python ui_phase_2.py  
Sending 6 packets  
Sent packet 0 with sequence number 0  
Sent packet 1 with sequence number 1  
Sent packet 2 with sequence number 0  
Sent packet 3 with sequence number 1  
Sent packet 4 with sequence number 0
```

Sent packet 5 with sequence number 1
File sent successfully!

Section 1(b): Below changes introduce bit errors randomly for testing purposes.

I can adjust the probability or the error handling mechanisms as needed.

Terminal 1 (Server):



```
def receive_file():
    for i in range(num_packets):
        packet, client_address = server_socket.recvfrom(2048) # Increased buffer size
        seq_num, checksum, packet_data = struct.unpack("III" + str(len(packet) - 8) + "s", packet)

        # Introduce bit error in data packet for testing
        if random.random() < 0.1: # 10% chance to introduce bit error
            packet_data = introduce_bit_error(packet_data)
```

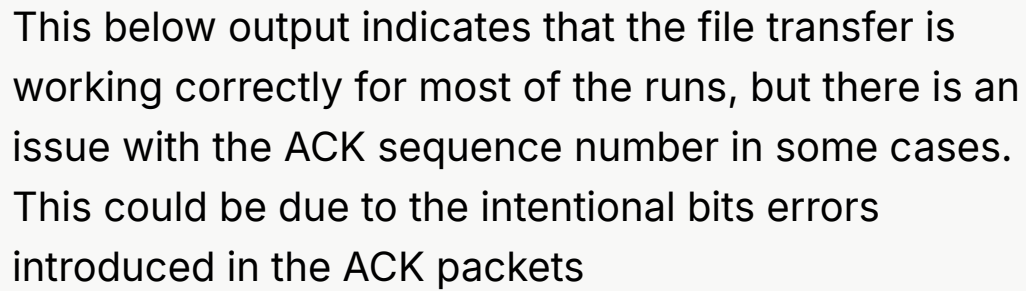
```
PS C:\Users\Vayush_Pandey\Dev\RD\> python ui_phase_2.py
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Error: Incorrect ACK sequence number. Expected 1, got 3
Sent packet 4 with sequence number 1
[]

PS C:\Users\Vayush_Pandey\Dev\RD\> python server_phase_2.py
Error: Incorrect sequence number
```

```
$ python server_phase_2.py
Error: Incorrect sequence number
```

```
$ python ui_phase_2.py
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 0
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 0
Error: Incorrect ACK sequence number. Expected 1, got 3
Sent packet 4 with sequence number 1
```

Terminal 1 (Performance plots):



The screenshot displays the Visual Studio Code interface during a network simulation. The Explorer sidebar on the left shows a project named 'RDT' containing files like '_pycache_', 'client_phase_2.py', 'client_rdt.py', 'Client.py', 'Design document Phase 1 File Transfer...', 'image.bmp', 'performance_plots.py', 'phase_2.jpg', 'ReadMe.txt', 'received_image.bmp', 'received_image.jpg', 'server_phase_2.py', 'server_rdt.py', 'Server.py', 'ui_phase_2.py', and 'ui.py'. The main editor area shows the source code of 'send_file' in 'client_phase_2.py', which includes logic for sending packets over a socket and handling acknowledgments. The integrated Terminal at the bottom shows the command 'python performance_plots.py' being executed, resulting in a series of messages indicating the successful transmission and reception of six packets between client and server.

```
PS C:\Users\Ayush_Pandey\Dev\RDT> python performance_plots.py
```

Sending 6 packets

Sent packet 0 with sequence number 0

Sent packet 1 with sequence number 1

Sent packet 2 with sequence number 0

Sent packet 3 with sequence number 1

Sent packet 4 with sequence number 0

Sent packet 5 with sequence number 1

File sent successfully!

File received successfully!

Sending 6 packets

Sent packet 0 with sequence number 0

Sent packet 1 with sequence number 1

Sent packet 2 with sequence number 0

Sent packet 3 with sequence number 1

Sent packet 4 with sequence number 0

Sent packet 5 with sequence number 1

File sent successfully!

File received successfully!

Sending 6 packets

Sent packet 0 with sequence number 0

Sent packet 1 with sequence number 1

Sent packet 2 with sequence number 0

Sent packet 3 with sequence number 1

Sent packet 4 with sequence number 0

Sent packet 5 with sequence number 1

File sent successfully!

File received successfully!

Sending 6 packets

Sent packet 0 with sequence number 0

Sent packet 1 with sequence number 1

Sent packet 2 with sequence number 0

Sent packet 3 with sequence number 1

Sent packet 4 with sequence number 0

Sent packet 5 with sequence number 1

File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1

File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1

File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1

File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1

File received successfully!
File sent successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1

File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1

File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!

File received successfully!
Sending 6 packets
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File received successfully!
File sent successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!

File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!

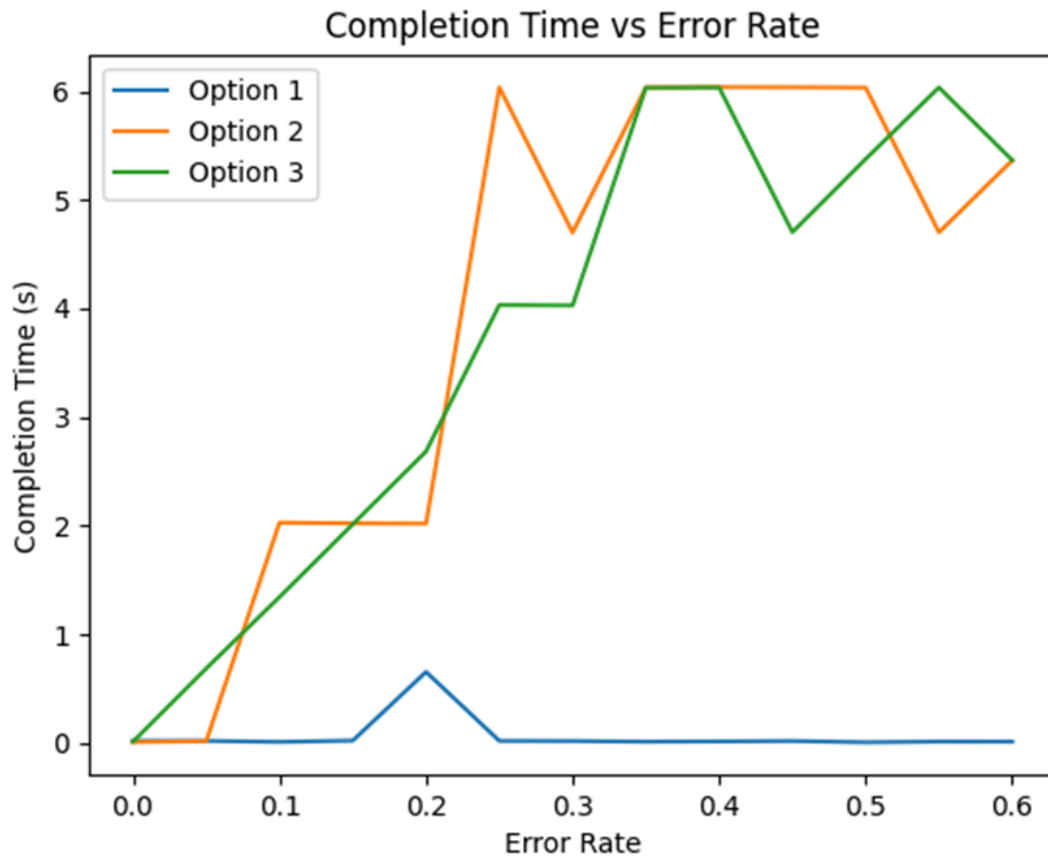
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1

Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0

Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!

Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0
Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Sent packet 1 with sequence number 1
Sent packet 2 with sequence number 0

Sent packet 3 with sequence number 1
Sent packet 4 with sequence number 0
Sent packet 5 with sequence number 1
Sent packet 5 with sequence number 1
File sent successfully!
File received successfully!
Sending 6 packets
Sent packet 0 with sequence number 0
Error: Incorrect ACK sequence number. Expected 0, got 2147483648
Sent packet 1 with sequence number 0



graph LR

A["Start: Pick a Transfer File (JPEG recommended)"] → B["Make Packet Function: Parse & Break into Packets (Fixed Size)"];


```

B → C("Use UDP Sockets for RDT2.2");
C → D{"Implement RDT Features:\n- Sequence Numbers\n- Checksum (Custom)\n- ACKs (RDT 2.2 NAK-free)"};
D → E("RDT2.2 Receiver: Assemble Packets in Order");
E → F("Deliver Entire File to Application");
F → G{Implement Data Transfer Scenarios:};
G → H1["Option 1: No Loss/Bit-Errors"];
G → H2["Option 2: ACK Packet Bit-Error\n(Implement Recovery)"];
G → H3["Option 3: Data Packet Bit-Error\n(Implement Recovery)"];
H1 → I("Performance Plots:\nCompletion Time vs Loss/Error (0-60% in 5%)\nFor All 3 Options\nUse >= 500KB File");
H2 → I;
H3 → I;
I → J["Note:\n- >= 500KB File\n- Average of Multiple Runs per Point\n- Remove Print/Debug for Time Measurement"];
J → K[End];

```



Plot charts for completion time for all your 3 options for transferring the same file at 0% loss/error to 60% loss/error in increments of 5%. Use at least a 500KB file for transmission.

```

PS C:\Users\Ayush_Pandey\Dev\RDT> python performance_plots.py
Option 1, Error Rate 0.0: 0.019988298416137695 seconds
Option 1, Error Rate 0.05: 0.018965959548950195 seconds
Option 1, Error Rate 0.1: 0.009654362996419271 seconds
Option 1, Error Rate 0.15: 0.021660804748535156 seconds
Option 1, Error Rate 0.2: 0.6542024612426758 seconds
Option 1, Error Rate 0.25: 0.01931452751159668 seconds
Option 1, Error Rate 0.3: 0.017635901769002277 seconds
Option 1, Error Rate 0.35: 0.01165469487508138 seconds
Option 1, Error Rate 0.4: 0.014321645100911459 seconds
Option 1, Error Rate 0.45: 0.017978986104329426 seconds
Option 1, Error Rate 0.5: 0.006320794423421224 seconds

```

Option 1, Error Rate 0.55: 0.012656450271606445 seconds
Option 1, Error Rate 0.6: 0.011975288391113281 seconds
Option 2, Error Rate 0.0: 0.008327881495157877 seconds
Option 2, Error Rate 0.05: 0.01662762959798177 seconds
Error: Incorrect ACK sequence number. Expected 1, got 129
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 8
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Option 2, Error Rate 0.1: 2.0281412601470947 seconds
Error: Incorrect ACK sequence number. Expected 0, got 32768
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 2, Error Rate 0.15: 2.0220643679300943 seconds
Error: Incorrect ACK sequence number. Expected 0, got 67108864
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c

losed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 8388608
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Option 2, Error Rate 0.2: 2.019318183263143 seconds
Error: Incorrect ACK sequence number. Expected 0, got 2097152
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 8
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Incorrect ACK sequence number. Expected 1, got 17
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c

losed by the remote host
Option 2, Error Rate 0.25: 6.036103963851929 seconds
Error: Incorrect ACK sequence number. Expected 1, got 1025
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 1073758208
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Incorrect ACK sequence number. Expected 1, got 1073741825
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Option 2, Error Rate 0.3: 4.700354973475139 seconds
Error: Incorrect ACK sequence number. Expected 0, got 16384
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK

Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 16785408
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 2048
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 2, Error Rate 0.35: 6.034603516260783 seconds
Error: Incorrect ACK sequence number. Expected 0, got 67108864
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0

Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 2147483648
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 16
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 2, Error Rate 0.4: 6.03875470161438 seconds
Error: Incorrect ACK sequence number. Expected 0, got 2
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c

losed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 4194304
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 262144
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Option 2, Error Rate 0.45: 6.037092129389445 seconds
Error: Incorrect ACK sequence number. Expected 0, got 1024
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 16777216

Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 2048
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 2, Error Rate 0.5: 6.034005085627238 seconds
Error: Incorrect ACK sequence number. Expected 0, got 16384
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 0, got 4456448
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK

Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 1, got 65537
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 2, Error Rate 0.55: 4.701932986577352 seconds
Error: Incorrect ACK sequence number. Expected 0, got 4194308
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 1, got 0
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Incorrect ACK sequence number. Expected 1, got 129
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for packet
Timeout waiting for ACK
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host

Error: Incorrect ACK sequence number. Expected 1, got 529
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for ACK
Timeout waiting for packet
Error: Incorrect sequence number. Expected 0, got 1
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 2, Error Rate 0.6: 5.36283016204834 seconds
Option 3, Error Rate 0.0: 0.011638800303141275 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Option 3, Error Rate 0.05: 0.686095396677653 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
Option 3, Error Rate 0.1: 1.3435686429341633 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Timeout waiting for ACK
Option 3, Error Rate 0.15: 2.0141589641571045 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK

ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 3, Error Rate 0.2: 2.6837806701660156 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 3, Error Rate 0.25: 4.032174030939738 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host

losed by the remote host
Option 3, Error Rate 0.3: 4.02835210164388 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Option 3, Error Rate 0.35: 6.0335110028584795 seconds
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c

losed by the remote host
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Timeout waiting for ACK
Option 3, Error Rate 0.4: 6.035097042719523 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly c
losed by the remote host
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK

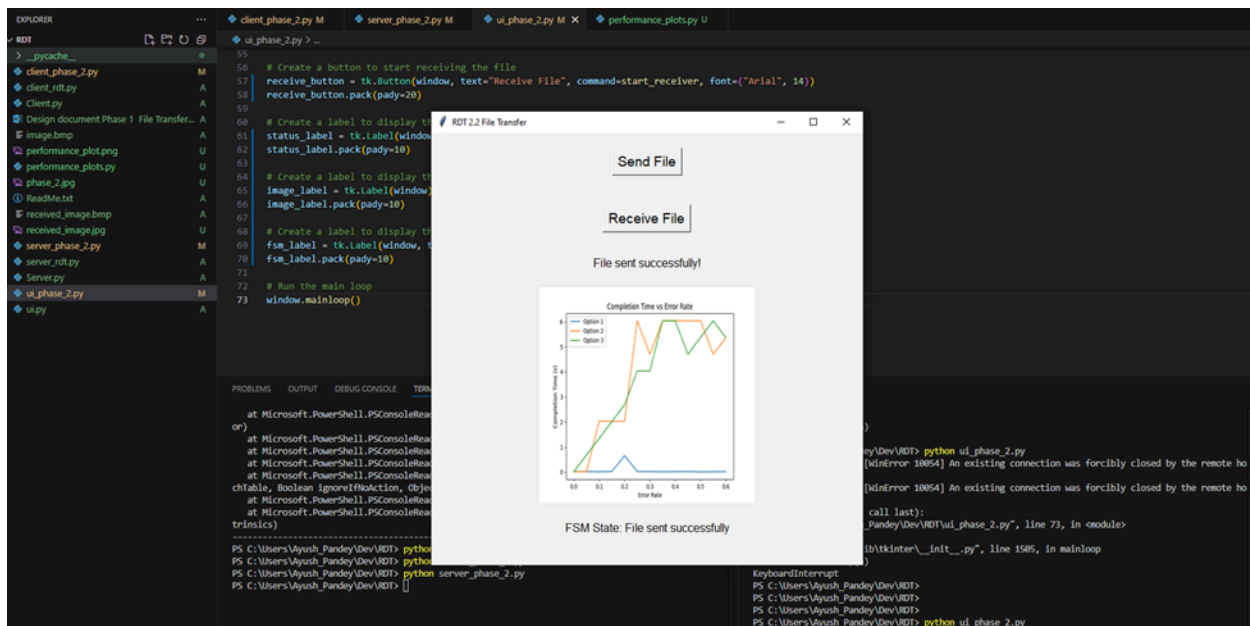
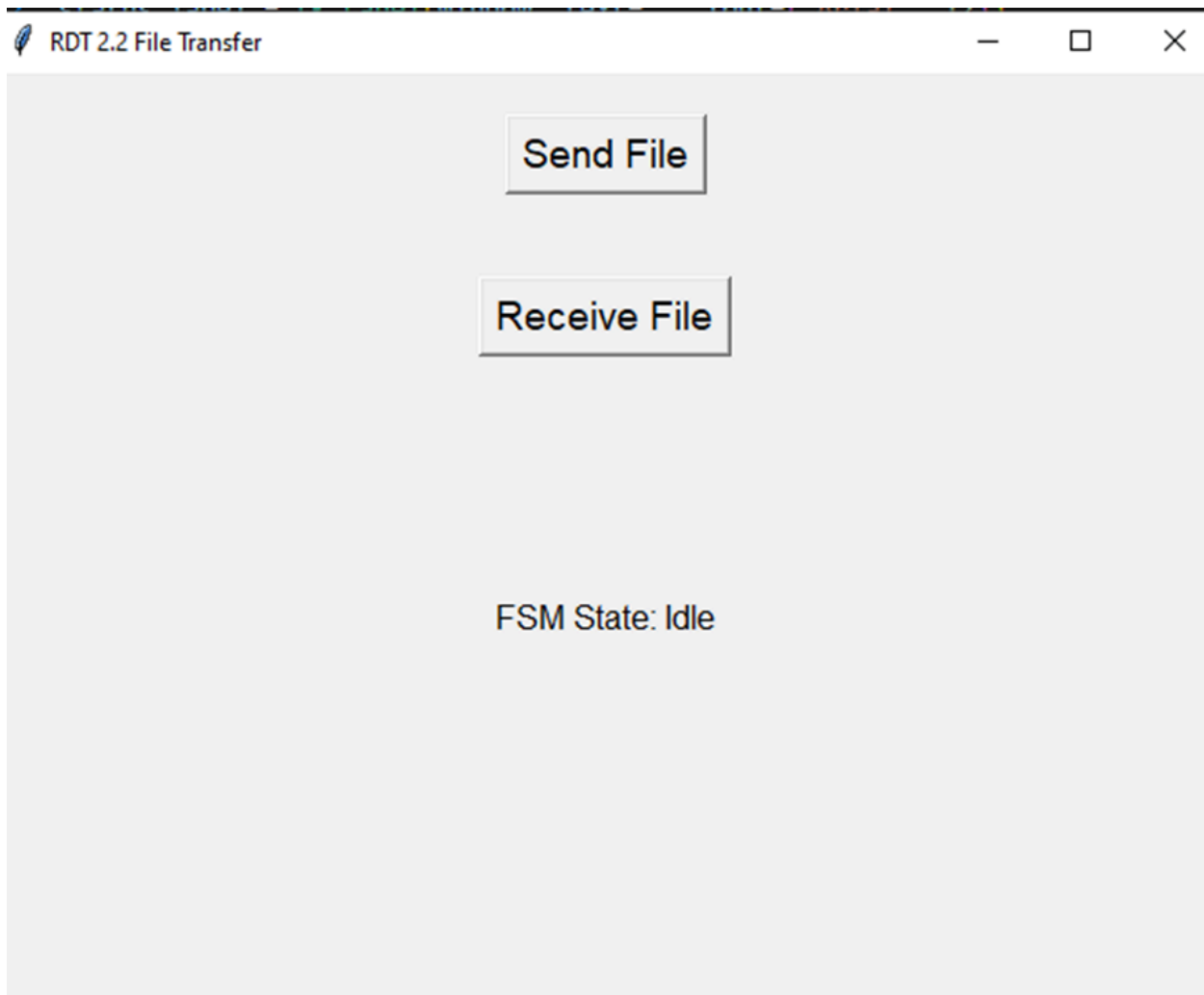
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 3, Error Rate 0.45: 4.702890078226726 seconds
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Timeout waiting for ACK
Option 3, Error Rate 0.5: 5.376731952031453 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for packet

Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 3, Error Rate 0.55: 6.0352442264556885 seconds
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Checksum mismatch

Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Error: Checksum mismatch
Timeout waiting for packet
Timeout waiting for ACK
Error: Checksum mismatch
Timeout waiting for ACK
Timeout waiting for packet
Error: Checksum mismatch
Timeout waiting for ACK
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host
Option 3, Error Rate 0.6: 5.365259965260823 seconds



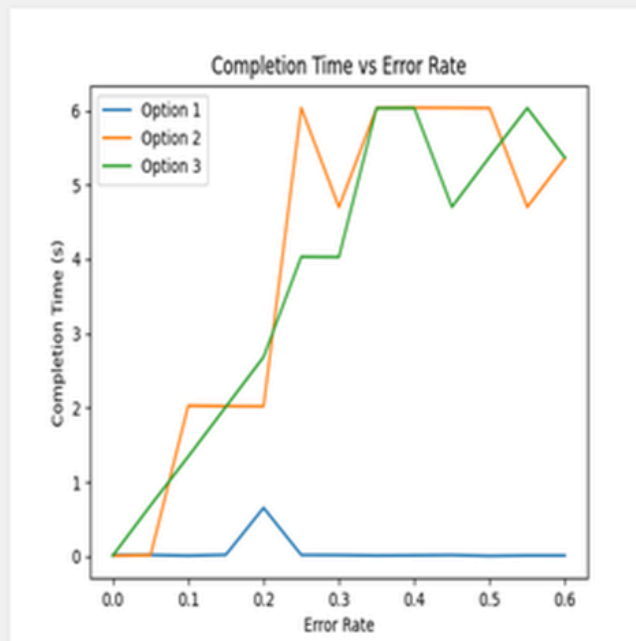
To create a high-quality GUI for the data transfer process, we can enhance the existing `ui_phase_2.py` script by adding features such as displaying the image being transferred, showing the sender and receiver FSM (Finite State Machine) states, and improving the overall look and feel of the GUI.



Send File

Receive File

File received successfully!



FSM State: File received successfully

```
PS C:\Users\Ayush_Pandey\Dev\RDT> python ui_phase_2.py
```

```
FSM State: Sent number of packets
```

```
Sent number of packets
```

```
FSM State: Sent packet 0 with sequence number 0
```

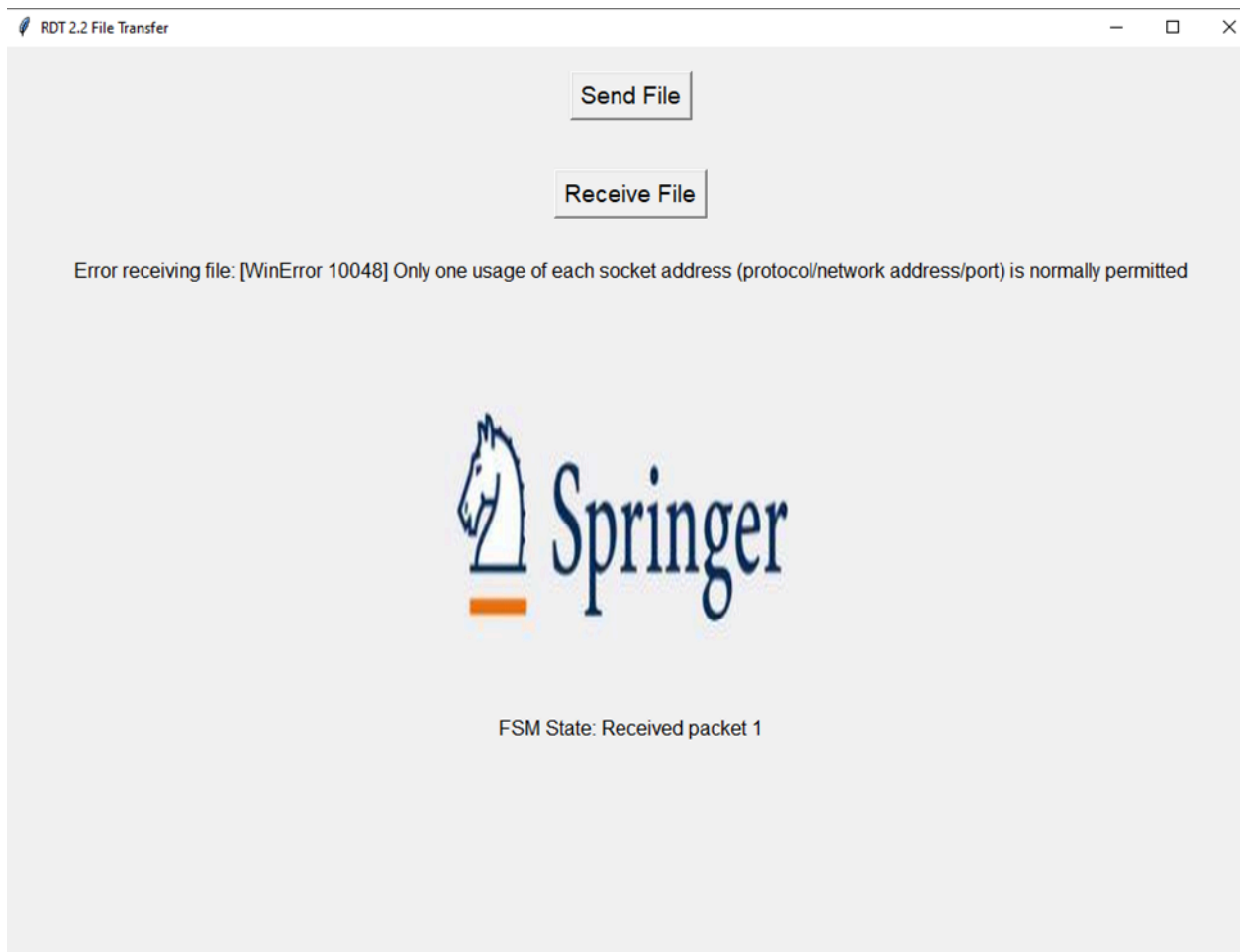
```
Sent packet 0 with sequence number 0
```

```
ConnectionResetError: [WinError 10054] An existing
```

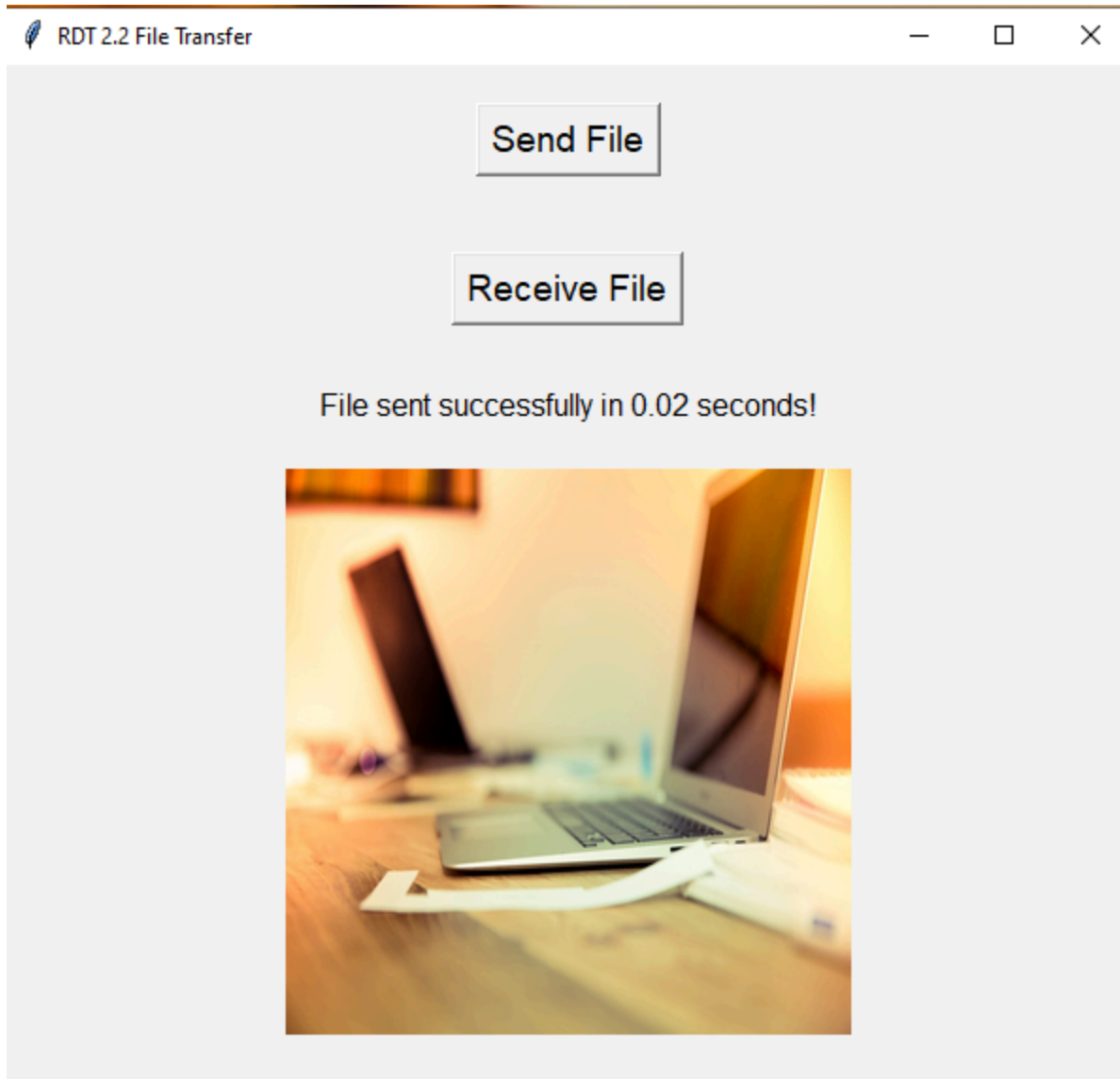
connection was forcibly closed by the remote host
FSM State: File sent successfully
File sent successfully
Server is ready to receive
Error receiving file: [WinError 10048] Only one usage of each socket address
(protocol/network address/port) is normally permitted
Error receiving file: [WinError 10048] Only one usage of each socket address
(protocol/network address/port) is normally permitted
Error receiving file: [WinError 10048] Only one usage of each socket address
(protocol/network address/port) is normally permitted
Error receiving file: [WinError 10048] Only one usage of each socket address
(protocol/network address/port) is normally permitted
FSM State: Sent number of packets
Sent number of packets
FSM State: Sent packet 0 with sequence number 0
Sent packet 0 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 1 with sequence number 0
Sent packet 1 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 2 with sequence number 0
Sent packet 2 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 3 with sequence number 0
Sent packet 3 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 4 with sequence number 0
Sent packet 4 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 5 with sequence number 0
Sent packet 5 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 6 with sequence number 0
Sent packet 6 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 7 with sequence number 0

Sent packet 7 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 8 with sequence number 0
Sent packet 8 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 9 with sequence number 0
Sent packet 9 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 10 with sequence number 0
Sent packet 10 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 11 with sequence number 0
Sent packet 11 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 12 with sequence number 0
Sent packet 12 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 13 with sequence number 0
Sent packet 13 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 14 with sequence number 0
Sent packet 14 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 15 with sequence number 0
Sent packet 15 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 16 with sequence number 0
Sent packet 16 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 17 with sequence number 0
Sent packet 17 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 18 with sequence number 0
Sent packet 18 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 19 with sequence number 0

Sent packet 19 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 20 with sequence number 0
Sent packet 20 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 21 with sequence number 0
Sent packet 21 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 22 with sequence number 0
Sent packet 22 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 23 with sequence number 0
Sent packet 23 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 24 with sequence number 0
Sent packet 24 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 25 with sequence number 0
Sent packet 25 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 26 with sequence number 0
Sent packet 26 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 27 with sequence number 0
Sent packet 27 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 28 with sequence number 0
Sent packet 28 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 29 with sequence number 0
Sent packet 29 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 30 with sequence number 0
Sent packet 30 with sequence number 0
Timeout waiting for ACK
FSM State: Sent packet 31 with sequence number 0



Use at least a 500KB file for transmission.



5. Future Improvements



For Extra Credit (if applicable), include the following points in your documentation.

- In Section 2.2, include a chart illustrating Total Transmission Time, Number of Retransmissions, and Number of Duplicate ACKs Received. Compare the impact of different delay ranges on these metrics to analyze performance variations.
- In Section 2.3, Compare the performance of checksum and CRC-16 based on error detection capability, computational overhead, and impact on retransmissions, and summarize the findings in a comparison table.
- Include the results in your document using a comparison table presenting metrics such as completion time, number of retransmissions, throughput, and CPU usage, along with a chart or figure (e.g., bar or line graph) to visually illustrate the performance differences between single-threaded and multi-threaded implementations; optionally, add screenshots of the output for reference.