Reliable Protocol Testing

Ashkan Zahedanaraki
Gagan Bahugun
December 12th, 2023

Features

1. Reading Commands

- Commands are read from stdin.
- An empty command (all whitespace) should not report an error.

2. Separating Commands

- Any non-empty read from stdin should be separated into commands.
- Supports only simple commands, defined as a single program name with optional arguments and optional I/O redirection.

3. Parsing Commands

- Commands are parsed into the following components:
    - Program to run
    - Arguments to the program
    - stdin redirection
    - stdout redirection
    - stderr redirection
- Redirection supports using ~ to expand to the user's home directory.

4. Executing Commands

- The PATH environment variable is used for locating and executing commands unless the command contains a /.

5. Exiting

- Upon exiting a command, the exit status is displayed.
- If a command cannot be found, an error message is shown.

6. UDP-Based Protocol

- Implementation of a reliable protocol using UDP.
- Supports both IPv4 and IPv6.

7. Sender Functionality

- Reads input from the keyboard and writes to a UDP socket.

- Takes command line arguments for the IP address of the proxy (or receiver) and the port.

8. Receiver Functionality

- Reads from a UDP socket and writes output to the console.
- Accepts a command line argument for the port to receive from.

9. Proxy Functionality

- Sits between the sender and receiver.
- Randomly drops data from the sender and acknowledgements from the receiver.
- Randomly delays packets from either the sender or receiver.
- Accepts command line arguments for IP address and ports for listening and sending and for probabilities of dropping and delaying packets.

10. Acknowledgements Handling

- The receiver sends an acknowledgement back to the sender via the proxy.
- The sender resends the packet if no acknowledgement is received within a reasonable time.

11. Graphical User Interface (GUI)

- A GUI to graph data related to packets sent and received by the sender, receiver, and proxy.
- The GUI may use TCP for its implementation.

12. Dynamic Configuration

- Ability to dynamically change the probabilities of dropping or delaying packets during runtime.

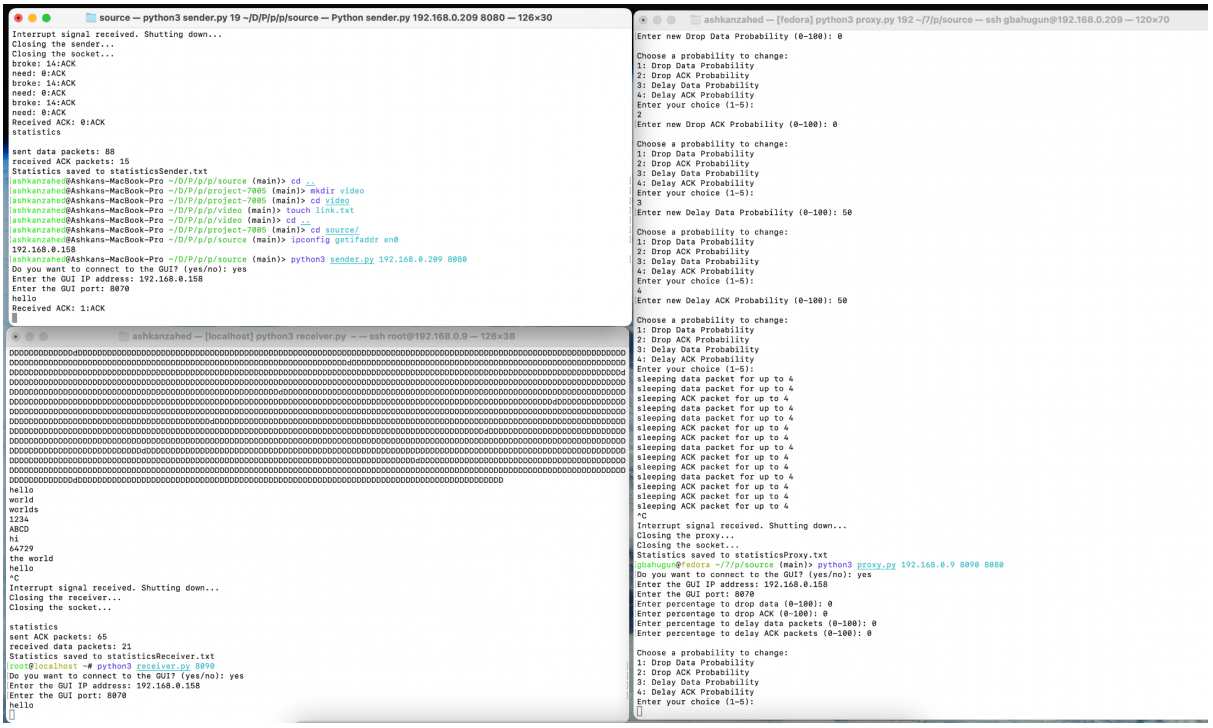
13. Statistics

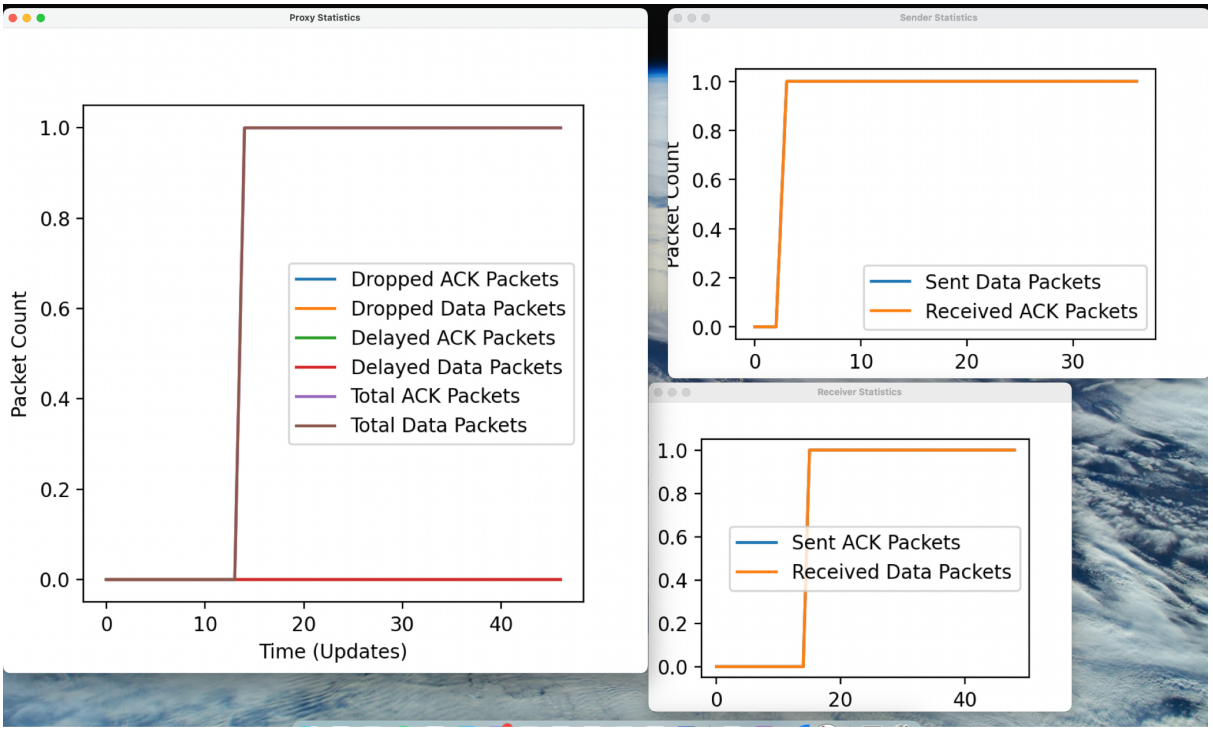
- All programs (sender, receiver, proxy) maintain and store statistics on the number of packets sent and received.
- Statistics are stored in a file for later analysis.

Test Results

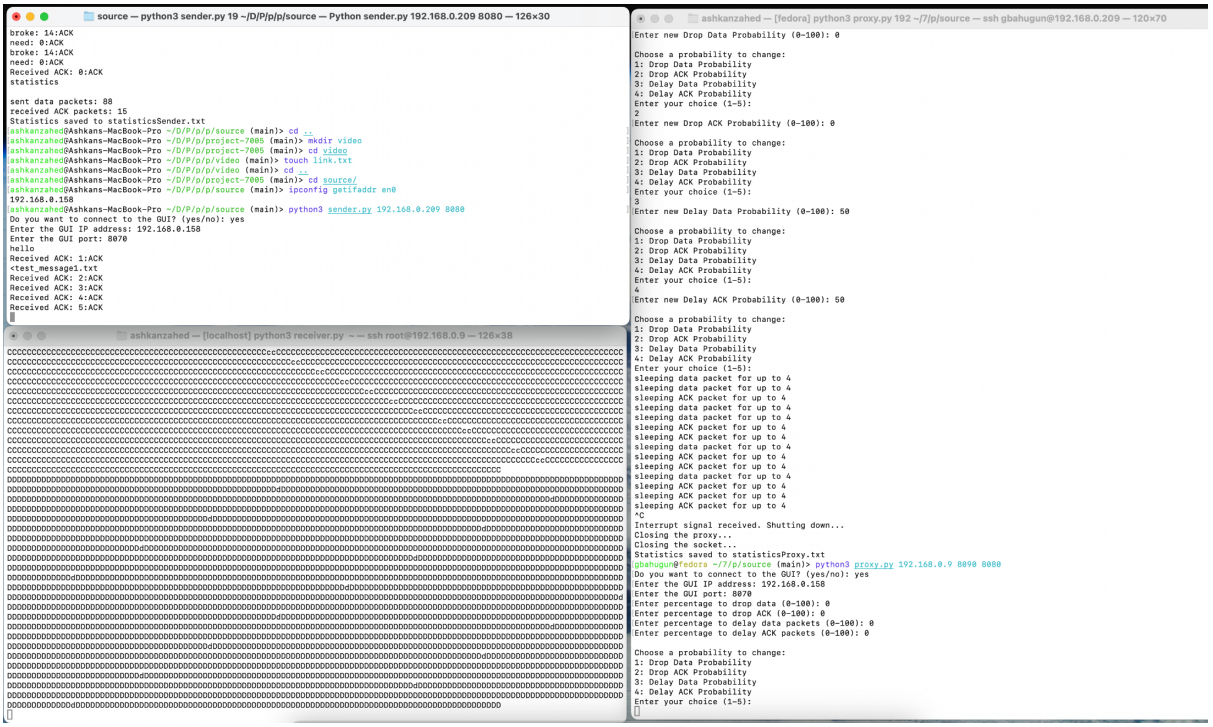
Description	Status	Example
Perfect network. No delays or drops of packets with keyboard input	Passed	Example 1
Perfect network. No delays or drops of packets with file input	Passed	Example 2
100% ACK packet delay	Passed	Example 3
100% Data packet delay	Passed	Example 4
100% ACK packet drop	Passed	Example 5
100% Data packet drop	Passed	Example 6
50% ACK and Data packet drop	Passed	Example 7
50% ACK and Data packet delay	Passed	Example 8
Dynamic change of probability of delay and drop in proxy	Passed	Example 9

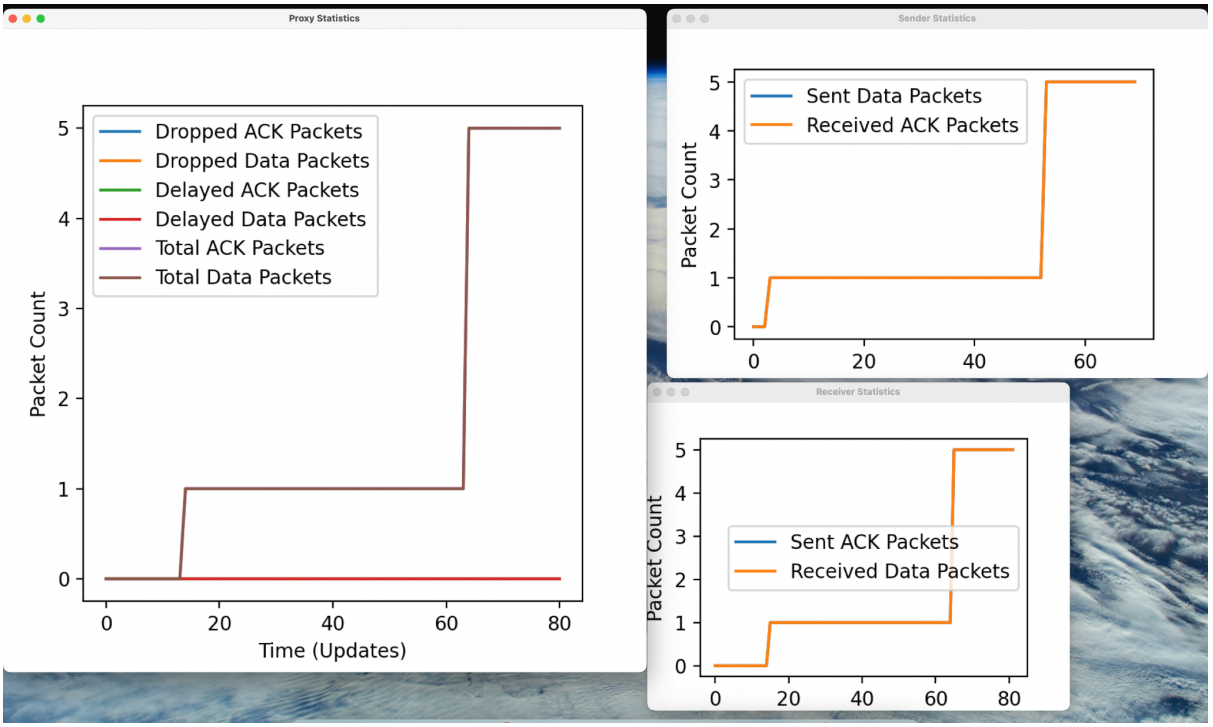
Perfect Network Keyboard Input



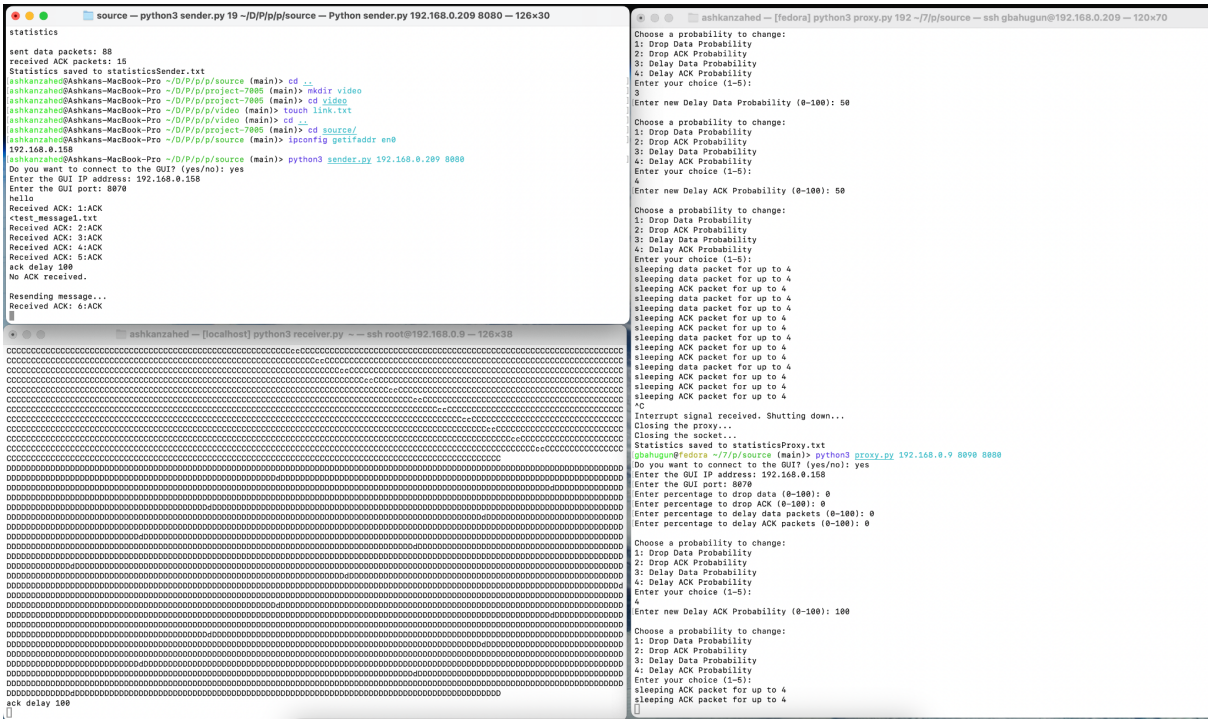


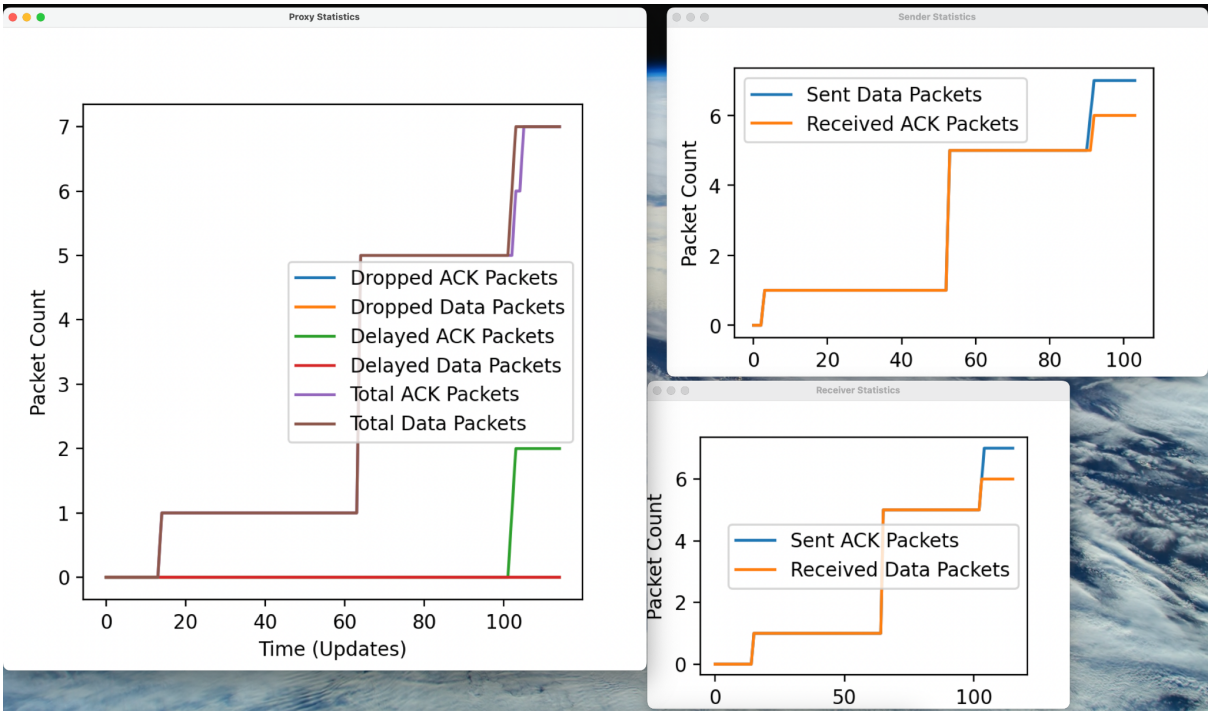
Perfect Network File Input



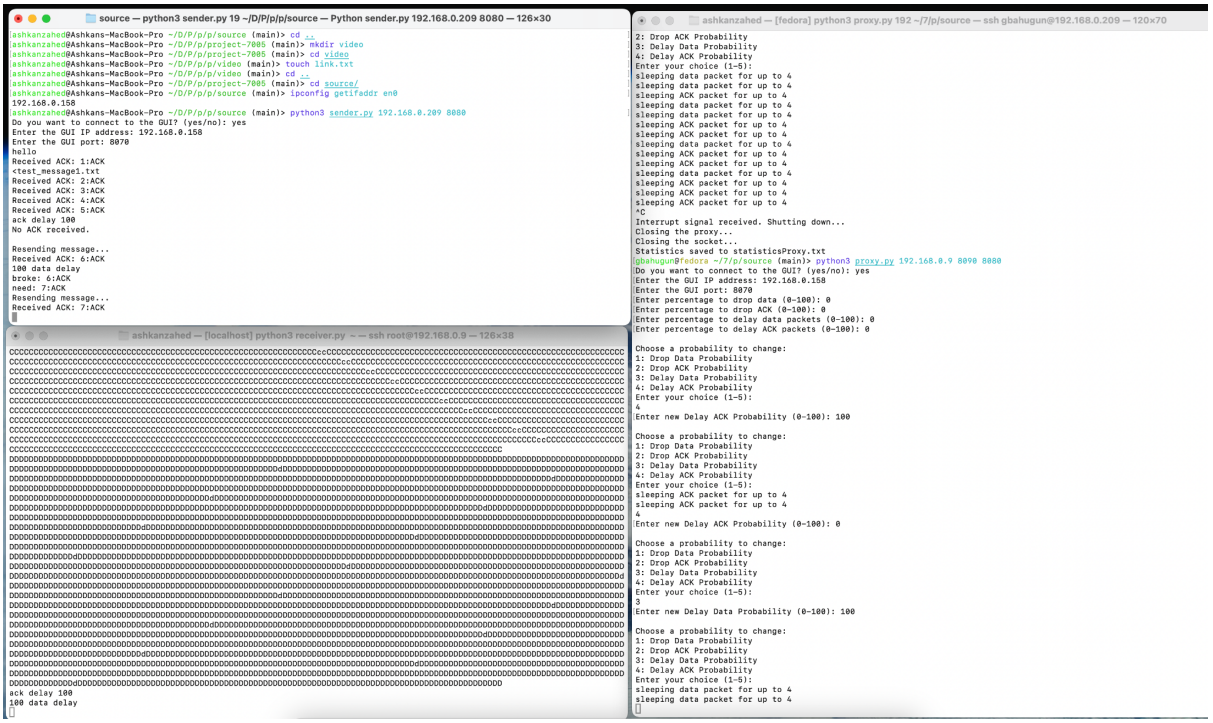


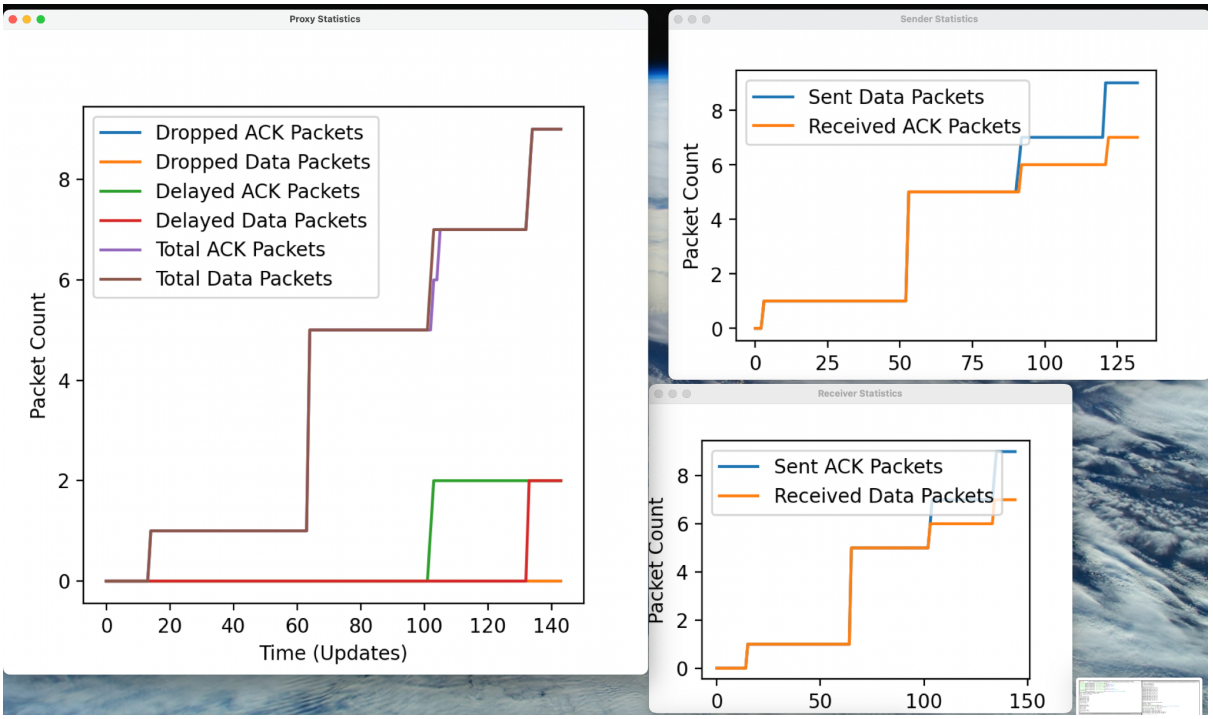
100% Ack Delay



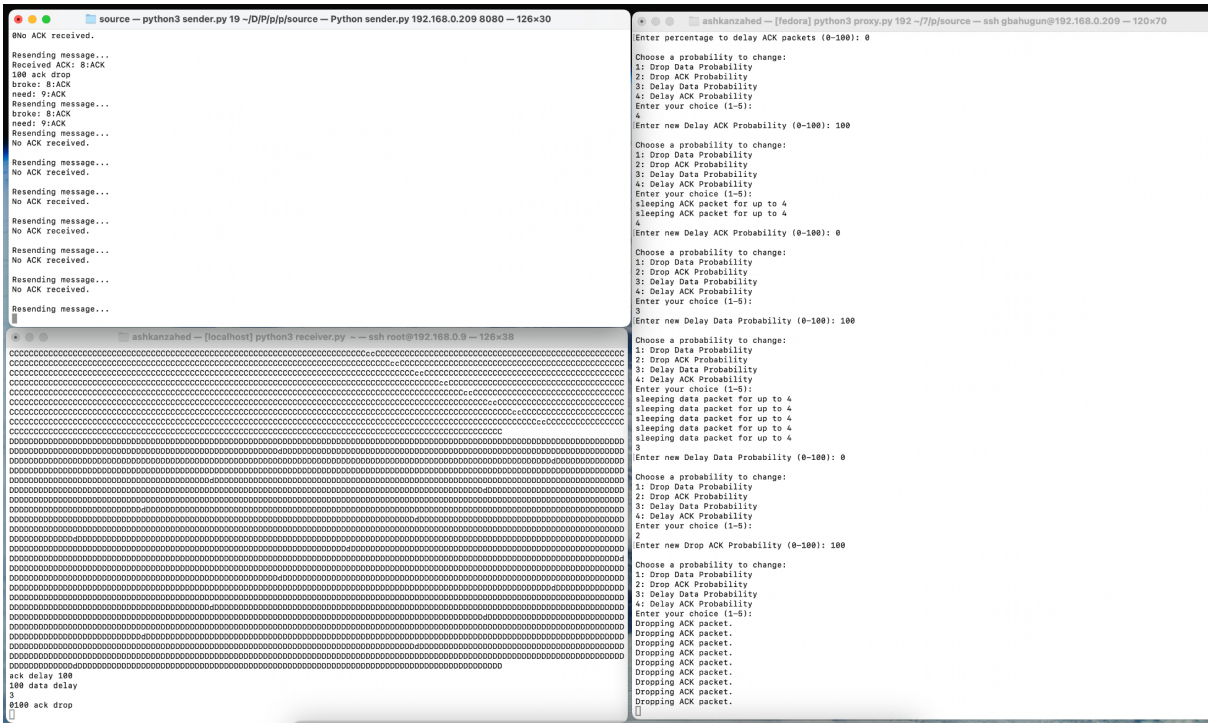


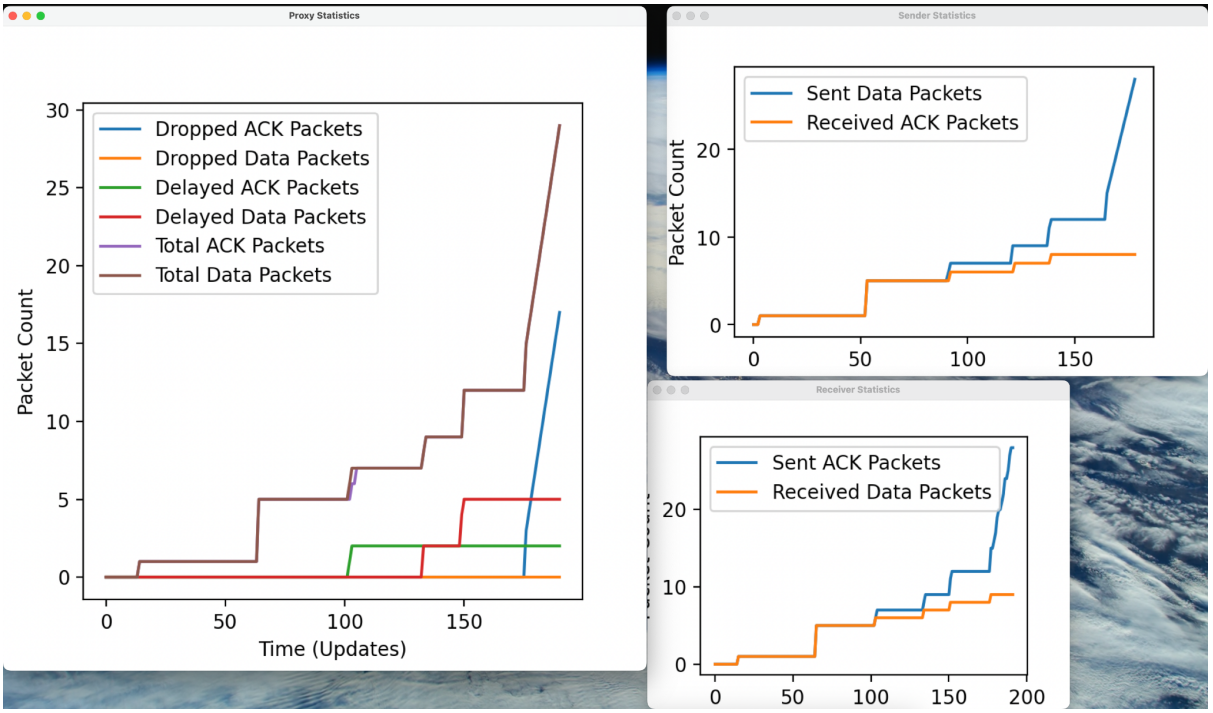
100% Data Delay



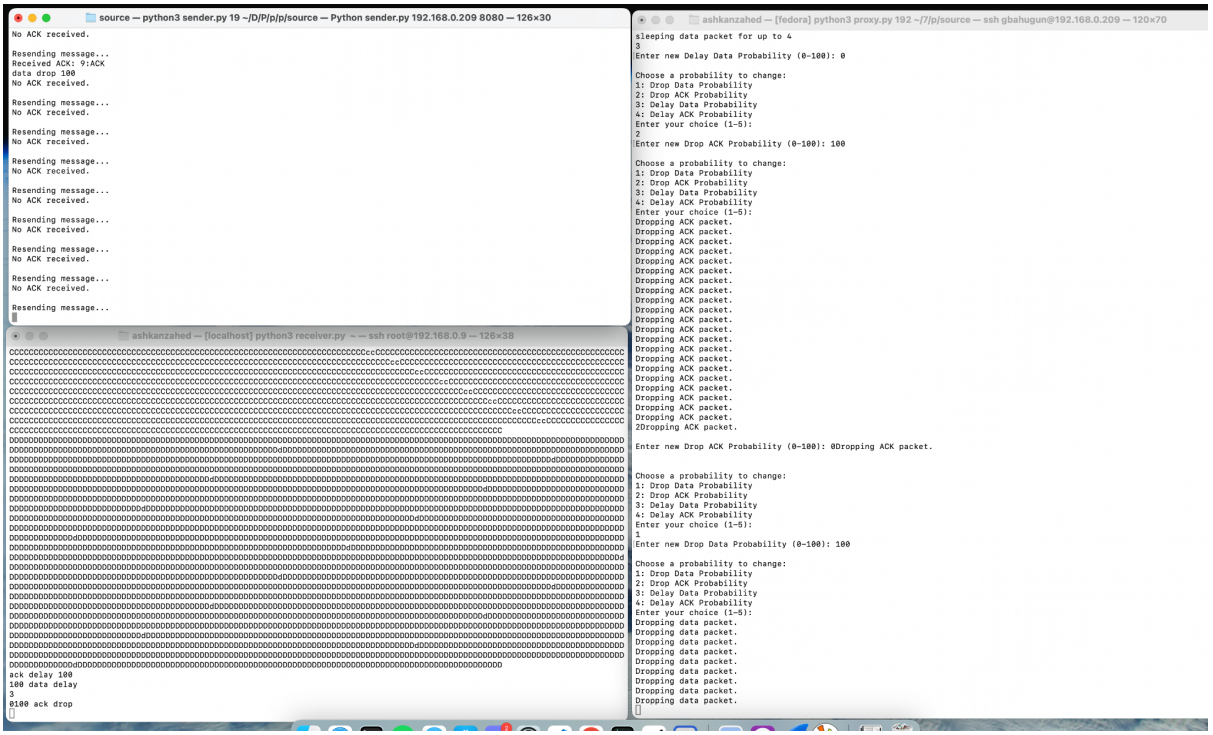


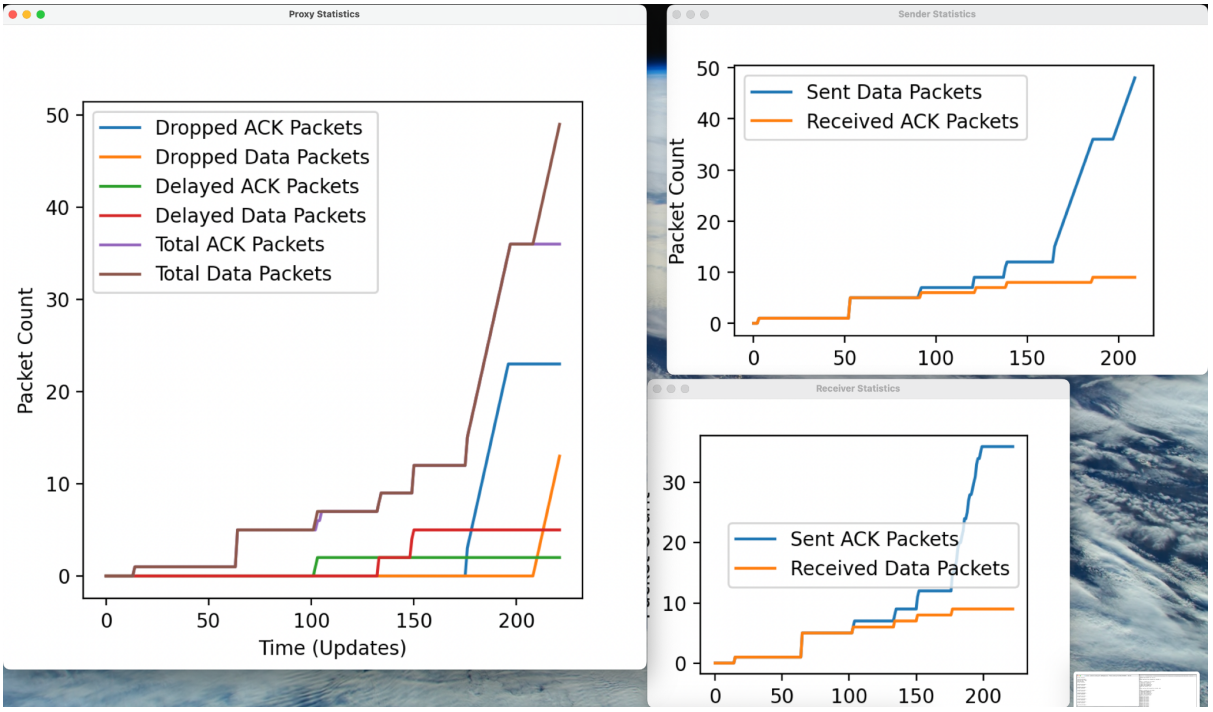
100% Ack Drop


source — python3 sender.py 19 ~/D/P/p/p/source — Python sender.py 192.168.0.209 8080 — 126×30
0No ACK received.
Resending message...
Received ACK: 8:ACK
100 ack drop
broke: 8:ACK
need: 9:ACK
Resending message...
broke: 8:ACK
need: 9:ACK
Resending message...
No ACK received.
Resending message...
No ACK received.
Resending message...
No ACK received.
Resending message...
No ACK received.
Resending message...
No ACK received.
Resending message...
No ACK received.
Resending message...
ashkanzahed — [localhost] python3 receiver.py — — ssh root@192.168.0.9 — 126×38
ack delay 100
100 data delay
3
0100 ack drop
ashkanzahed — [fedora] python3 proxy.py 192 ~/7/p/source — ssh gbahugun@192.168.0.209 — 120×70
Enter percentage to delay ACK packets (0-100): 0
Choose a probability to change:
1: Drop Data Probability
2: Drop ACK Probability
3: Delay Data Probability
4: Delay ACK Probability
Enter your choice (1-5):
4
Enter new Delay ACK Probability (0-100): 100
Choose a probability to change:
1: Drop Data Probability
2: Drop ACK Probability
3: Delay Data Probability
4: Delay ACK Probability
Enter your choice (1-5):
sleeping ACK packet for up to 4
sleeping ACK packet for up to 4
4
Enter new Delay ACK Probability (0-100): 0
Choose a probability to change:
1: Drop Data Probability
2: Drop ACK Probability
3: Delay Data Probability
4: Delay ACK Probability
Enter your choice (1-5):
3
Enter new Delay Data Probability (0-100): 100
Choose a probability to change:
1: Drop Data Probability
2: Drop ACK Probability
3: Delay Data Probability
4: Delay ACK Probability
Enter your choice (1-5):
sleeping data packet for up to 4
sleeping data packet for up to 4
sleeping data packet for up to 4
sleeping data packet for up to 4
sleeping data packet for up to 4
3
Enter new Delay Data Probability (0-100): 0
Choose a probability to change:
1: Drop Data Probability
2: Drop ACK Probability
3: Delay Data Probability
4: Delay ACK Probability
Enter your choice (1-5):
2
Enter new Drop ACK Probability (0-100): 100
Choose a probability to change:
1: Drop Data Probability
2: Drop ACK Probability
3: Delay Data Probability
4: Delay ACK Probability
Enter your choice (1-5):
Dropping ACK packet.
Dropping ACK packet.
Dropping ACK packet.
Dropping ACK packet.
Dropping ACK packet.
Dropping ACK packet.
Dropping ACK packet.
Dropping ACK packet.
Dropping ACK packet.

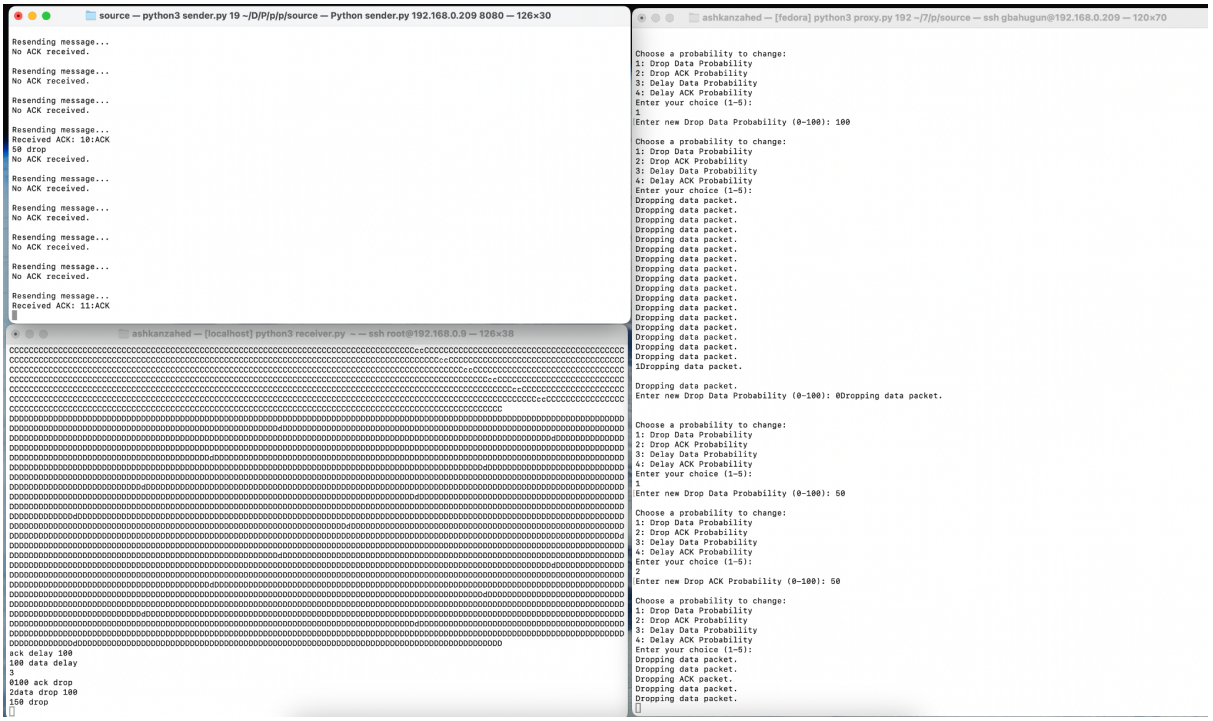

Proxy Statistics
Dropped ACK Packets
Dropped Data Packets
Delayed ACK Packets
Delayed Data Packets
Total ACK Packets
Total Data Packets
Packet Count
Time (Updates)
Sender Statistics
Sent Data Packets
Received ACK Packets
Packet Count
Receiver Statistics
Sent ACK Packets
Received Data Packets

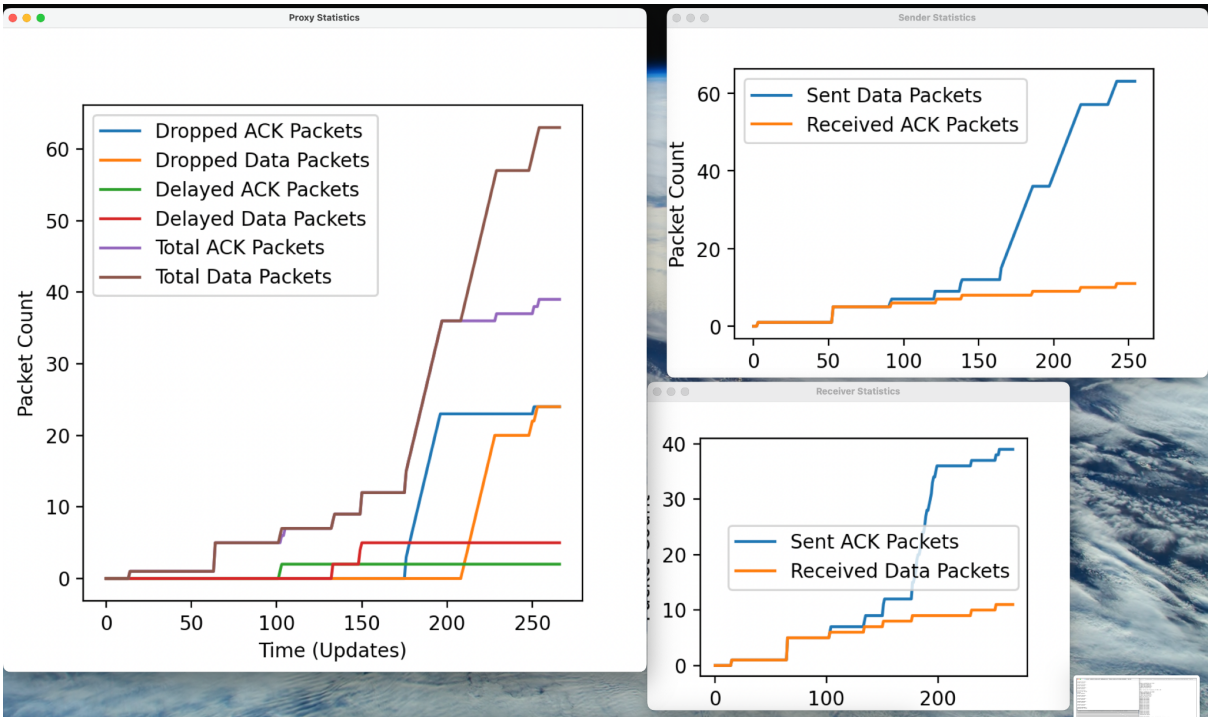
100% Data drop



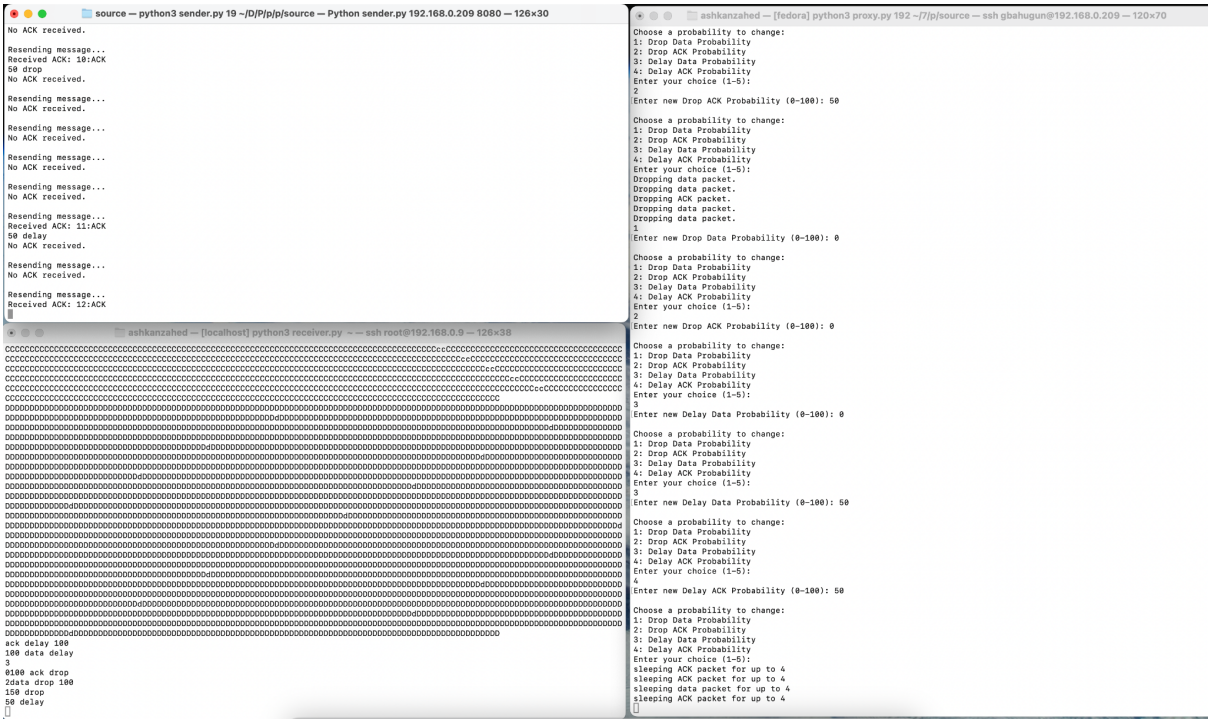


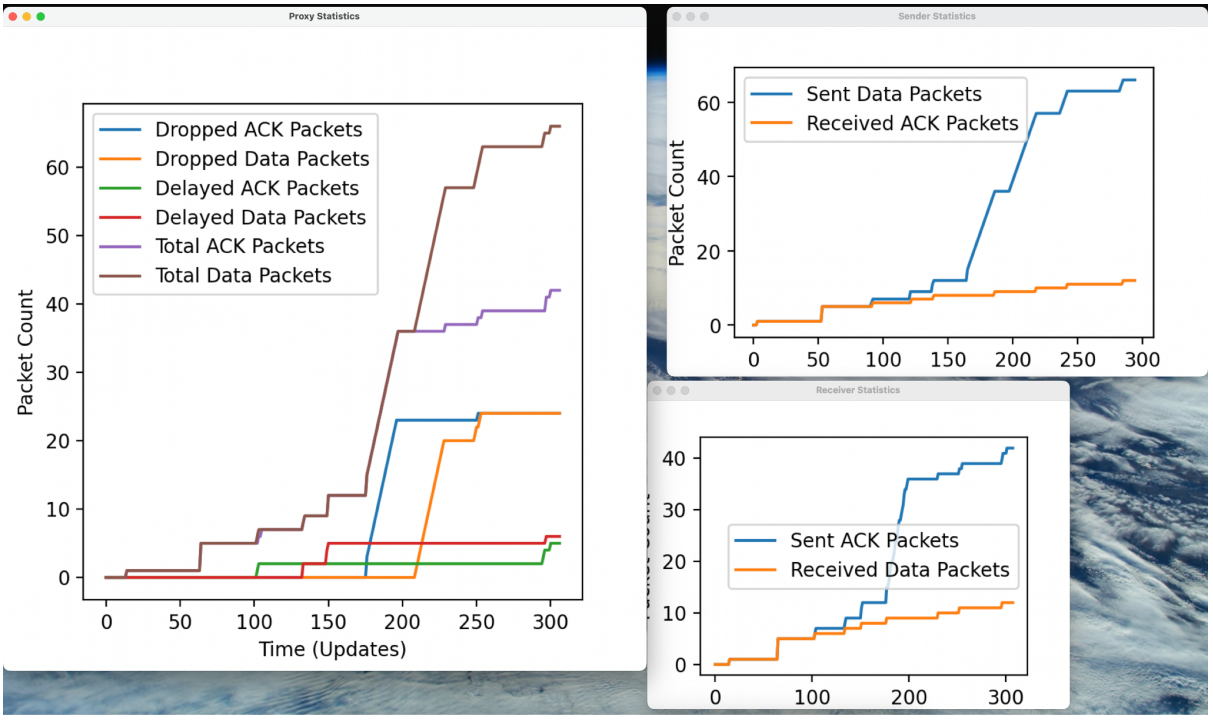
50% Ack and Data Drop





50% Ack and Data Delay





Dynamic Proxy Drop and Delay Change