

The image shows three terminal windows running on a macOS system. The left window, titled 'Ethan_Iannicelli_hw3 -- Python server.py -- 66x53', shows the server's output. It starts by waiting for a file, then receives packets 0 through 9. It handles out-of-order packets and packet corruption by ignoring them and re-sending ACKs. It also simulates packet loss and corruption. The middle window, titled 'Ethan_Iannicelli_hw3 -- Python intermediary.py -- 69x53', shows the intermediary's output. It starts by listening on port 12346, then simulates packet loss and corruption. The right window, titled 'Ethan_Iannicelli_hw3 -- zsh -- 66x53', shows the client's output. It starts by entering the filename 'file_to_send_long.txt', then sends packets 0 through 9. It handles timeouts and retransmissions. At the bottom of the right window, the user runs the command 'diff file_to_send_long.txt received_files/file_to_send_long.txt' and the output shows no difference between the files.

```
ethaniannicelli@Ethan's-MacBook-Air-2 Ethan_Iannicelli_hw3 % python3 server.py
Waiting for file (ctrl-C to quit)...
Received in-order packet 0, sending ACK.
Received in-order packet 1, sending ACK.
Packet corrupted! Ignoring.
Received previously acked packet 1, re-sending ACK.
Received in-order packet 2, sending ACK.
Out-of-order packet 4 received. Expecting 3. Ignoring.
Received previously acked packet 2, re-sending ACK.
Out-of-order packet 4 received. Expecting 3. Ignoring.
Out-of-order packet 5 received. Expecting 3. Ignoring.
Out-of-order packet 6 received. Expecting 3. Ignoring.
Received in-order packet 3, sending ACK.
Received in-order packet 4, sending ACK.
Received in-order packet 5, sending ACK.
Received in-order packet 6, sending ACK.
Received in-order packet 7, sending ACK.
Packet corrupted! Ignoring.
Received in-order packet 8, sending ACK.
Received in-order packet 9, sending ACK.
File saved to received_files/file_to_send_long.txt
Waiting for file (ctrl-C to quit)...
```

```
ethaniannicelli@Ethan's-MacBook-Air-2 Ethan_Iannicelli_hw3 % python3 intermediary.py
Intermediary started, listening on port ('127.0.0.1', 12346) (ctrl-C to quit)...
---Simulating packet loss.
---Simulating packet loss.
---Simulating packet corruption.
---Simulating packet loss.
---Simulating packet loss.
---Simulating packet corruption.
---Simulating packet corruption.
```

```
ethaniannicelli@Ethan's-MacBook-Air-2 Ethan_Iannicelli_hw3 % python3 client.py
Enter the filename to send (enter 'quit' to stop the program):
file_to_send_long.txt
Sent packet 0
ACK received for 0
Sent packet 1
Sent packet 2
Sent packet 3
Sent packet 4
Timeout! Retransmitting unacknowledged packets.
Retransmitted packet 1
Retransmitted packet 2
Retransmitted packet 3
Retransmitted packet 4
ACK received for 1
Sent packet 5
Timeout! Retransmitting unacknowledged packets.
Retransmitted packet 2
Retransmitted packet 3
Retransmitted packet 4
Retransmitted packet 5
ACK received for 2
Sent packet 6
Timeout! Retransmitting unacknowledged packets.
Retransmitted packet 3
Retransmitted packet 4
Retransmitted packet 5
Retransmitted packet 6
ACK received for 3
Sent packet 7
ACK received for 4
Sent packet 8
ACK received for 5
ACK received for 6
ACK received for 7
Timeout! Retransmitting unacknowledged packets.
Retransmitted packet 8
Timeout! Retransmitting unacknowledged packets.
Retransmitted packet 8
ACK received for 8
Sent packet 9
ACK received for 9
File file_to_send_long.txt sent successfully!
Enter the filename to send (enter 'quit' to stop the program):
quit
ethaniannicelli@Ethan's-MacBook-Air-2 Ethan_Iannicelli_hw3 % diff file_to_send_long.txt received_files/file_to_send_long.txt
ethaniannicelli@Ethan's-MacBook-Air-2 Ethan_Iannicelli_hw3 % diff file_to_send_long.txt received_files/file_to_send_long.txt
ethaniannicelli@Ethan's-MacBook-Air-2 Ethan_Iannicelli_hw3 %
```

The above shows the output for all three programs running at the same time. The left panel is the server, the middle panel is the intermediary, and the right panel is the client. In the middle, each '-' represents a packet that was sent without disruption. The left shows how packets are handled when obstruction occurs, and the right shows the client side of this interaction. At the bottom of the right panel, I ran the diff command to show the output of comparing the 'send' file and the 'received' file - There is no difference between them