Computer Networks
Project#2: Stop and wait

Reem Mahmoud	4525
Mai Hossam	4387
Mariam Hassan	4242
Lougina Ehab	4350

Introduction:

Stop and wait is used in connection-oriented communication. It offers error and control flow and is used in Link and Transport layers.

Suppose you've a file and you want to send this file from one side to the other(server to client), you will need to split the file into chunks of data of fixed length and add the data of one chunk to a UDP datagram packet in the data field of the packet. This is implemented as :

The server sends a single datagram, and blocks until an acknowledgment from the client is received (or until a timeout expires).

Notes about the project:

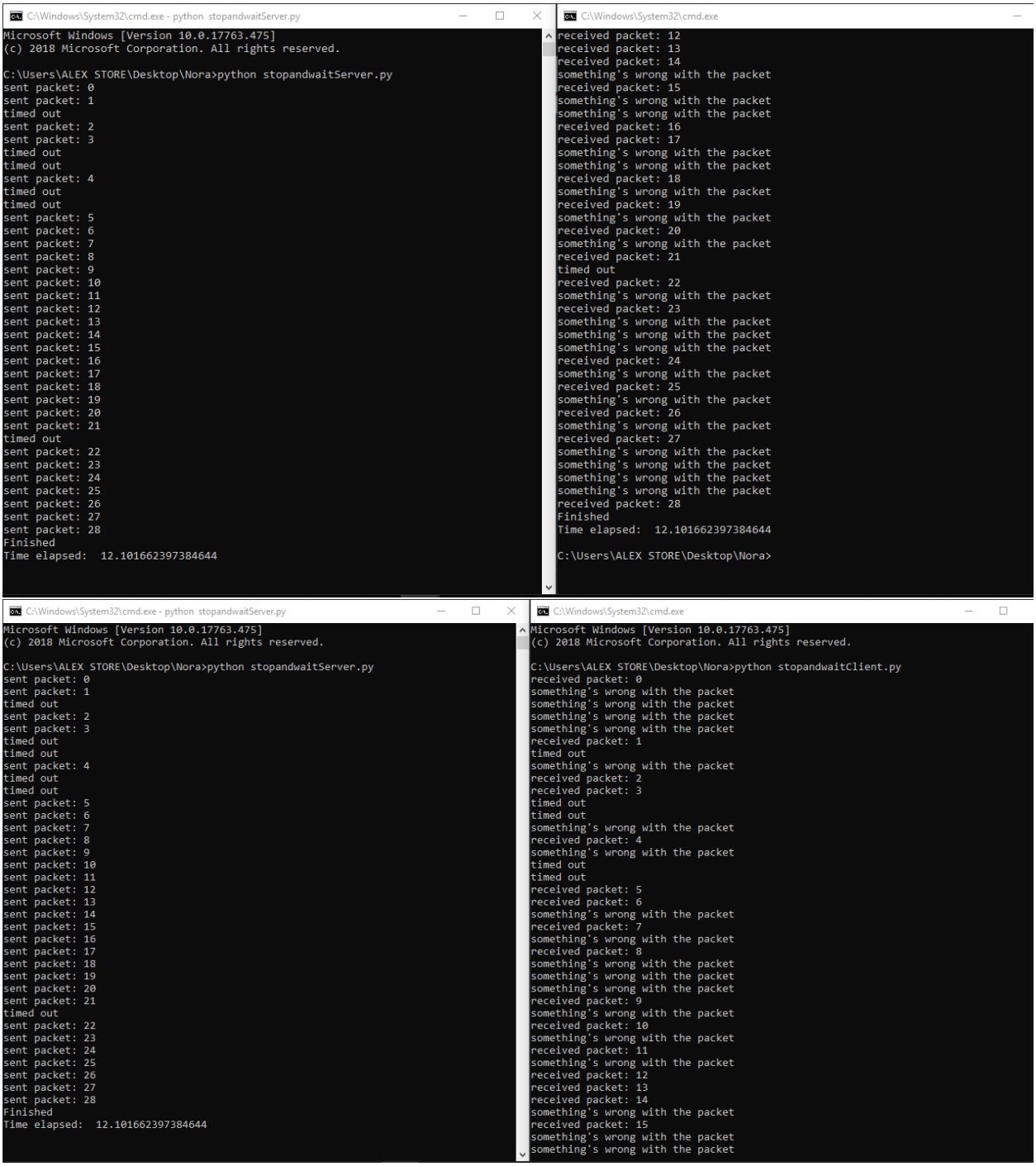
Project was fully coded in python. It consists of two .py files: "stopandwaitServer.py" and "stopandwaitClient.py" and two text files: "file.txt" and "output.txt".

Workflow between server and client:

1. Sockets are created for both the server and client.
2. The client sends a datagram to the server giving the filename for the transfer. This send is backed up by a timeout in case the datagram is lost.
3. The server forks off a child process to handle the client.
4. The server searches for and opens the file "file.txt" and calculates it's length.
5. The server sends the size of the file to the client in a string.
6. A timer starts at the server side. This handles timeout events.
7. The server (child) creates a UDP socket to handle file transfer to the client.
8. Server sends its first datagram, the server uses some random number generator random() function to decide with probability p if the datagram would be passed to the method send() or just ignore sending it
9. Whenever a datagram arrives, an ACK is sent out by the client to the server.
10. If you choose to discard the package and not to send it from the server the timer will end at the server waiting for the ACK that it will never come from the client (since the packet wasn't sent to it) and the packet will be resent again from the server.

11. Window is updated and datagrams are ordered at the client side.
12. Steps are repeated until the whole file is sent and no other datagrams remains.
13. The file received at the client side is called “output.txt”
14. Connection is closed.

Program in action:



Sources referenced:

- https://www.geeksforgeeks.org/stop-and-wait-arq/?fbclid=IwAR2zGvP8gvTHq57_9Is4JQiFmL8AR3WldidllbwPyqALfElJ-AeQseeYiRo
- https://stackoverflow.com/questions/15909064/python-implementation-for-stop-and-wait-algorithm?fbclid=IwAR1COxFV3I-RL-cs5a6tU0eVkk1JriefMs7UD4fdUFzregz-ibVuEmc-ohE
- https://github.com/mj2266/stop-and-wait-protocol?fbclid=IwAR2goOmzvOSn5FUWzUTRu4GHFsEjFslhv1Axh7FXy_0hRciga6UdKbQKBiA