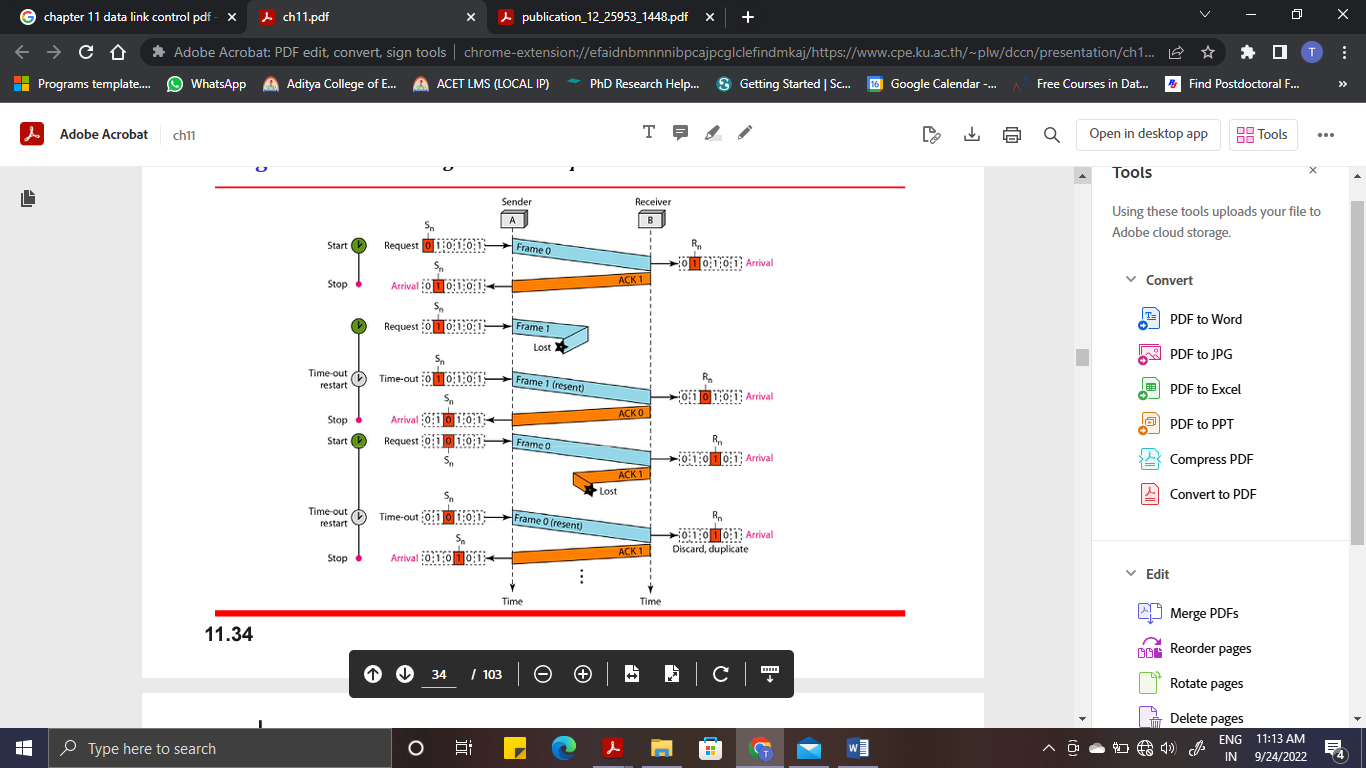
8. Write a Program to implement Stop and Wait Protocol.



**Implementation at Sender side ( server.py):**

**import socket  
import time  
import random  
s=socket.socket()  
s.bind(("localhost", 8020))  
s.listen(5)  
c, adr = s.accept()  
print("connection to " + str(adr) + " established")  
a=int(input("enter total number of frames"))  
x = 0  
print("sending -->", x)  
c.send(str(x).encode())  
while( a > 1 ):  
 timer = 5  
 t=random.randint(1,7)  
 msg = c.recv(1).decode()  
 if( timer > t):  
 time.sleep(3)  
 print("ack-->", msg)  
 x=int(msg)  
 print("sending -->", str(x))  
 c.send(str(x).encode())  
 else:  
 time.sleep(3)  
 print("timeout")  
 print("sending again-->", x)  
 c.send(str(x).encode())  
 a=a+1  
 a = a-1**

**Implementation at Receiver side ( client.py):**

**import socket  
import time  
s=socket.socket()  
s.connect(("localhost", 8020))  
while(1):  
 msg=s.recv(1).decode()  
 print("Received --> ", msg)  
 x=int(msg)  
 if(x==0):  
 x=x+1  
 s.send(str(x).encode())  
 else:  
 x=x-1  
 s.send(str(x).encode())**

**Output:**

**Note: first run server.py and then client.py**

**SERVER SIDE:**

**connection to ('127.0.0.1', 55894) established**

**enter total number of frames 5**

**sending --> 0**

**ack--> 1**

**sending --> 1**

**timeout**

**sending again--> 1**

**timeout**

**sending again--> 1**

**timeout**

**sending again--> 1**

**ack--> 0**

**sending --> 0**

**ack--> 1**

**sending --> 1**

**timeout**

**sending again--> 1**

**timeout**

**sending again--> 1**

**timeout**

**sending again--> 1**

**ack--> 0**

**sending --> 0**

**Process finished with exit code 0**

**CLIENT SIDE:**

**Received --> 0**

**Received --> 1**

**Received --> 1**

**Received --> 1**

**Received --> 1**

**Received --> 0**

**Received --> 1**

**Received --> 1**

**Received --> 1**

**Received --> 1**

**Received --> 0**

**Process finished with exit code 1**