|  |  |
| --- | --- |
| **Name: Vinit Shah** | **SAP ID:60004220097** |
| **Date of Performance:28-08-24** | **Date of Submission:12-09-24** |

**Experiment No: 4**

**Aim:** Write a program to implement Sliding Window techniques.

**Theory:**

## GO-N-BACK ARQ

import random

import time

window\_size= int(input("Number of frames that can be sent before needing an ACK:"))

total\_frames= int(input ("Total number of frames to send:"))

timeout= int(input("Timeout in seconds:"))

# Simulate sending a frame

def send\_frame(frame\_number):

print(f"Sending frame {frame\_number}")

return random.choice([True, False]) # Simulate a frame loss (False) or success (True)

# Simulate receiving an ACK

def receive\_ack(expected\_frame):

time.sleep(1) # Simulate delay

return random.choice([expected\_frame, None]) # Simulate ACK loss (None) or success (expected\_frame)

# Go-Back-N ARQ simulation

def go\_back\_n\_arq():

base = 0 # First frame in the window

next\_frame\_to\_send = 0 # Next frame to send

ack\_received = -1 # Last acknowledged frame

while base < total\_frames:

while next\_frame\_to\_send < base + window\_size and next\_frame\_to\_send < total\_frames:

if send\_frame(next\_frame\_to\_send):

print(f"Frame {next\_frame\_to\_send} sent successfully.")

else:

print(f"Frame {next\_frame\_to\_send} lost.")

next\_frame\_to\_send += 1

ack = receive\_ack(base)

if ack is not None:

print(f"Received ACK for frame {ack}")

base = ack + 1

else:

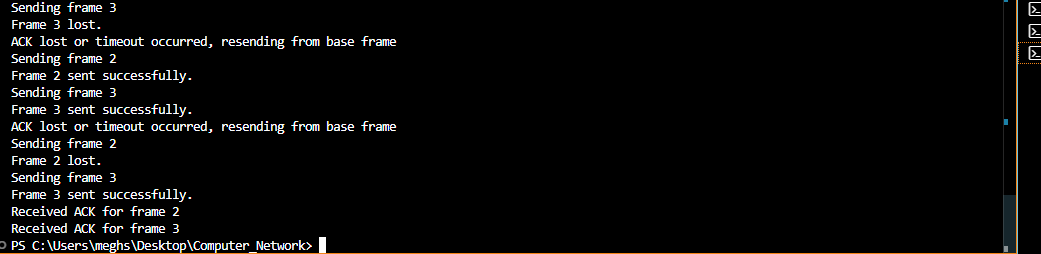
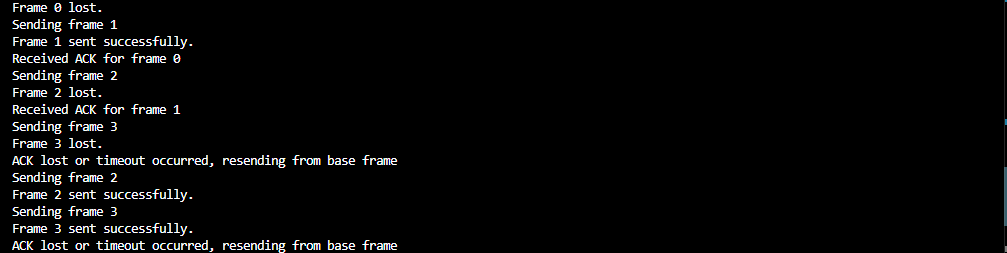
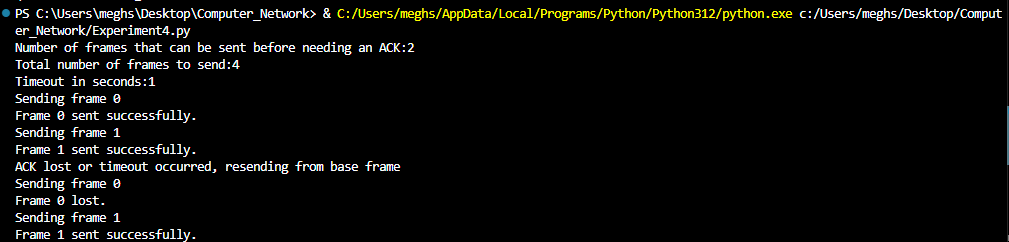
print("ACK lost or timeout occurred, resending from base frame")

next\_frame\_to\_send = base # Go back to the base frame and resend

if \_\_name\_\_ == "\_\_main\_\_":

go\_back\_n\_arq()

**Screenshots:**



**Conclusion:**

Thus, we have successfully studied and implemented sliding window approaches.