



Assignment 1

Go-Back-N Protocol

Computer Networks

Name	ID
Omar Mohamed Diaaeldin Ibrahim Elsayed	1802932
Ahmed Magdy Fahmy Mohamed Ahmed	1805862
Ahmed Magdi Mostafa Hosni	1808714
Ahmed Ayman Abd-Alaziz Sharf	1806171
Yousef Adel Ismail Mohammed Shalaby	1802267

Assignment 1

Sender.py:

Function: Send data in packets to receiver

A. Ideal State:

- **Code:**

```
import time, socket, sys, pickle
soc = socket.socket()
host = '127.0.0.1'
port = 1122
soc.bind((host, port))
soc.listen(1)
conn, addr = soc.accept()
while True:
    message = input(str("enter message :"))
    #message = 'Hey Buddy , How do you do ?'
    window_in = len(message)
    conn.send(message.encode())
    print(message.encode())
    messagelength = str(len(message))
    conn.send(messagelength.encode())
    packet_to_send_to_receiver = 0
    window_size_c = input("Do you want to define manuall window size ? Y/N ")
    if window_size_c.lower() == "y":
        window_size = int(input("enter size of window packet: "))
    else:
        window_size = window_in
    buffer = ""
    window_size = window_size - 1
    messagelength = int(messagelength)
    end_of_window = window_size
    while packet_to_send_to_receiver != messagelength:
        while packet_to_send_to_receiver != (messagelength - window_size):
            conn.send(message[packet_to_send_to_receiver].encode())
            print('Sending packet number {}, which is {}'.format(packet_to_send_to_receiver,
                                                                    message[packet_to_send_to_receiver]))

            buffer = conn.recv(1024)
            buffer = buffer.decode()
            if buffer != "ACK Lost":
                print('{} is received, sliding window is in the range of {} to {}'.format(buffer, str(
                    packet_to_send_to_receiver + 1), str(end_of_window + 1)))
                packet_to_send_to_receiver = packet_to_send_to_receiver + 1
                end_of_window = end_of_window + 1
            else:
                print('Ack is lost, sliding window is in the range of {} to {}'.format(
                    str(packet_to_send_to_receiver + 1),
                    , str(end_of_window + 1)))
```

```

while packet_to_send_to_receiver != messagelength:
    conn.send(message[packet_to_send_to_receiver].encode())
    print('Sending packet number {}, which is {}'.format(packet_to_send_to_receiver,
                                                         message[packet_to_send_to_receiver]))

    buffer = conn.recv(1024)
    buffer = buffer.decode()
    if buffer != "ACK Lost":
        print('{} is received, sliding window is in the range of {} to {}'.format(buffer, str(
            packet_to_send_to_receiver + 1), str(end_of_window + 1)))
        packet_to_send_to_receiver = packet_to_send_to_receiver + 1
    else:
        print('Ack is lost, sliding window is in the range of {} to {}'.format(
            str(packet_to_send_to_receiver + 1), str(end_of_window + 1)))
    exit_cond = input("Do you want to send another message ? Y/N")
    if exit_cond.lower() != "y":
        break

```

- **Screenshot of output at ideal Output:**

The image displays two screenshots of a Python IDE (likely PyCharm) showing the execution of a sliding window protocol. The top screenshot shows the initial setup and the first few packets being sent and received. The bottom screenshot shows the continuation of the process with more packets being sent and received, demonstrating the sliding window mechanism.

Top Screenshot:

```

1 import time, socket, sys, pickle
2
3 soc = socket.socket()
4 host = '127.0.0.1'
5 port = 1122
6 soc.bind((host, port))
7
8 soc.listen(1)
9 conn, addr = soc.accept()
10
11 while True:
12     message = input(str("enter message :"))
13     #message = 'Hey Buddy , How do you do ?'
14     window_ln = len(message)
15     conn.send(message.encode())
16     print(message.encode())

```

Run: sender receiver

```

"C:\Users\Omar Diaa\AppData\Local\Programs\Python\Python38-32\python.exe" "E:/Faculty/Senior - 1/First Term/Computer Networks/Assignments/Go-back-N-main/sender.py"
enter message :Omar Mohamed Diaaeldin Ibrahim
b'Omar Mohamed Diaaeldin Ibrahim'
Do you want to define manuvall window size ? Y/N Y
enter size of window packet: 2
Sending packet number 0, which is 0
ACK 0 is received, sliding window is in the range of 1 to 2
Sending packet number 1, which is m
ACK 1 is received, sliding window is in the range of 2 to 3
Sending packet number 2, which is a

```

Bottom Screenshot:

```

1 import time, socket, sys, pickle
2
3 soc = socket.socket()
4 host = '127.0.0.1'
5 port = 1122
6 soc.bind((host, port))
7
8 soc.listen(1)
9 conn, addr = soc.accept()
10
11 while True:
12     message = input(str("enter message :"))
13     #message = 'Hey Buddy , How do you do ?'
14     window_ln = len(message)
15     conn.send(message.encode())
16     print(message.encode())

```

Run: sender receiver

```

Sending packet number 11, which is d
ACK 11 is received, sliding window is in the range of 12 to 13
Sending packet number 12, which is
ACK 12 is received, sliding window is in the range of 13 to 14
Sending packet number 13, which is D
ACK 13 is received, sliding window is in the range of 14 to 15
Sending packet number 14, which is i
ACK 14 is received, sliding window is in the range of 15 to 16
Sending packet number 15, which is a
ACK 15 is received, sliding window is in the range of 16 to 17

```

Figure 1:Output at ideal State

B. Packets Lost:

• Code:

```
import time, socket, sys, pickle
soc = socket.socket()
host = '127.0.0.1'
port = 1122
soc.bind((host, port))
soc.listen(1)
conn, addr = soc.accept()
while True:
    message = input(str("enter message :"))
    #message = 'Hey Buddy , How do you do ?'
    window_in = len(message)
    conn.send(message.encode())
    print(message.encode())
    messagelength = str(len(message))
    conn.send(messagelength.encode())
    packet_to_send_to_receiver = 0
    window_size_c = input("Do you want to define manuall window size ? Y/N ")
    if window_size_c.lower() == "y":
        window_size = int(input("enter size of window packet: "))
    else:
        window_size = window_in
    buffer = ""
    window_size = window_size - 1
    messagelength = int(messagelength)
    end_of_window = window_size
    while packet_to_send_to_receiver != messagelength:
        while packet_to_send_to_receiver != (messagelength - window_size):
            conn.send(message[packet_to_send_to_receiver].encode())
            print('Sending packet number {}, which is {}'.format(packet_to_send_to_receiver,
                                                                    message[packet_to_send_to_receiver]))

            buffer = conn.recv(1024)
            buffer = buffer.decode()
            if buffer != "ACK Lost":
                print('{} is received, sliding window is in the range of {} to {}'.format(buffer, str(
                    packet_to_send_to_receiver + 1), str(end_of_window + 1)))
                packet_to_send_to_receiver = packet_to_send_to_receiver + 1
                end_of_window = end_of_window + 1
            else:
                print('Ack is lost, sliding window is in the range of {} to {}'.format(
                    str(packet_to_send_to_receiver + 1), str(end_of_window + 1)))
        while packet_to_send_to_receiver != messagelength:
            conn.send(message[packet_to_send_to_receiver].encode())
            print('Sending packet number {}, which is {}'.format(packet_to_send_to_receiver,
                                                                    message[packet_to_send_to_receiver]))

            buffer = conn.recv(1024)
            buffer = buffer.decode()
            if buffer != "ACK Lost":
                print('{} is received, sliding window is in the range of {} to {}'.format(buffer, str(
```

```

        packet_to_send_to_receiver + 1), str(end_of_window + 1)))
    packet_to_send_to_receiver = packet_to_send_to_receiver + 1
else:
    print('Ack is lost, sliding window is in the range of {} to {}'.format(
        str(packet_to_send_to_receiver + 1)
        , str(end_of_window + 1)))
exit_cond = input("Do you want to send another message ? Y/N")
if exit_cond.lower() != "y":
    break

```

• Screenshot of output at packets lost

The figure consists of two screenshots of a Python IDE, likely PyCharm, showing the execution of a Go-Back-N protocol implementation.

Top Screenshot: The code editor shows the sender's code (sender.py) and the receiver's code (receiver.py). The sender's code includes a while loop that sends packets and handles lost acknowledgments. The receiver's code includes a while loop that receives packets and handles lost acknowledgments. The Run console shows the output of the sender's code, including the message "Hey Buddy... How do you do?", the window size, and the packet numbers sent (0 and 1). The output also shows that the acknowledgment is lost, and the sliding window is updated.

Bottom Screenshot: The code editor shows the receiver's code (receiver.py). The Run console shows the output of the receiver's code, including the message "ACK 24 is received, sliding window is in the range of 25 to 26" and the packet numbers received (24 and 25). The output also shows that the acknowledgment is lost, and the sliding window is updated.

Figure 2: Output When Some Packets are lost

Receiver.py:

Function: Receive data in packets from Sender

A. Ideal State:

- Code:

```
import time, socket, sys, pickle
import random
s = socket.socket()
host = '127.0.0.1'
port = 1122
s.connect((host, port))
print("Connected...\n")
while True:
    m = s.recv(1024)

    m = m.decode()

    k = s.recv(1024)

    k = k.decode()
    k = int(k)
    i = 0
    a = ""
    Ack = ""
    f = random.randint(0, 1)
    message = ""
    while i != k:
        f = random.randint(0, 10)
        f = 1
        if (f == 6):
            Ack = "ACK Lost"
            print(f"Packet No {i} is lost, requesting it again")
            message = s.recv(1024)
            message = message.decode()
            s.send(Ack.encode())

        elif (f == 1):

            Ack = "ACK " + str(i)
            print(f"{Ack} is received.")
            message = s.recv(1024)
            message = message.decode()

            s.send(Ack.encode())

            a = a + message

            i = i + 1

    print("The message received is :", a)
```

- **Screenshot of output at ideal Output:**

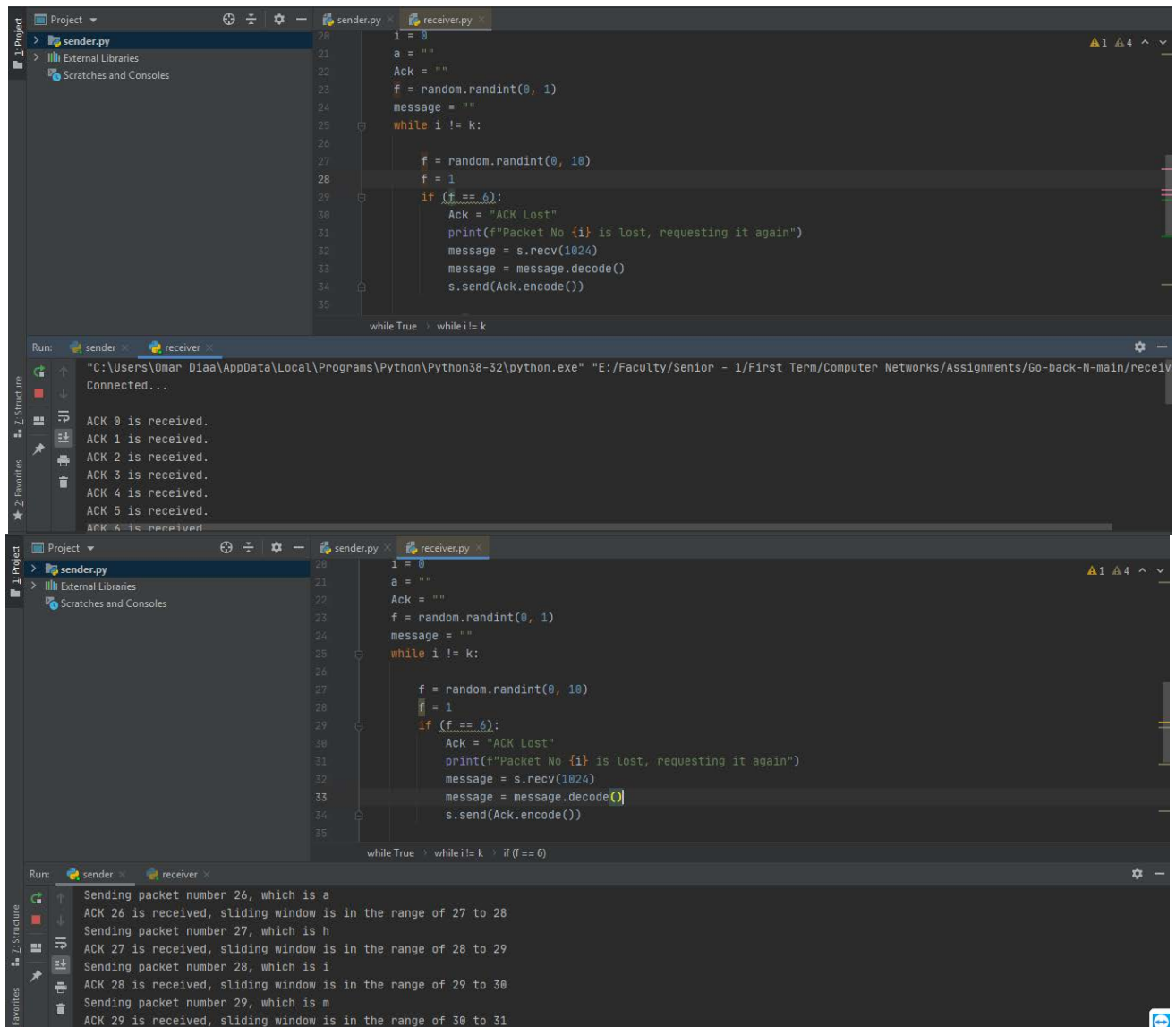


Figure 3: Output at ideal State

B. Packets Lost:

- **Code:**

```
import time, socket, sys, pickle
import random
s = socket.socket()
host = '127.0.0.1'
port = 1122
s.connect((host, port))
print("Connected...\n")
while True:
    m = s.recv(1024)

    m = m.decode()

    k = s.recv(1024)
    k = k.decode()
```

```

k = int(k)
i = 0
a = ""
Ack = ""
f = random.randint(0, 1)
message = ""
while i != k:
    f = random.randint(0, 10)
    #f = 1
    if (f == 6):
        Ack = "ACK Lost"
        print(f"Packet No {i} is lost, requesting it again")
        message = s.recv(1024)
        message = message.decode()
        s.send(Ack.encode())

    elif (f == 1):

        Ack = "ACK " + str(i)
        print(f"{Ack} is received.")
        message = s.recv(1024)
        message = message.decode()

        s.send(Ack.encode())

    a = a + message

    i = i + 1

print("The message received is :", a)

```

• Screenshot of output at packets lost

The screenshot displays a Python IDE with two files: `sender.py` and `receiver.py`. The `receiver.py` file contains the following code:

```

1 = 0
a = ""
Ack = ""
f = random.randint(0, 1)
message = ""
while i != k:
    f = random.randint(0, 10)
    #f = 1
    if (f == 6):
        Ack = "ACK Lost"
        print(f"Packet No {i} is lost, requesting it again")
        message = s.recv(1024)
        message = message.decode()
        s.send(Ack.encode())

    elif (f == 1):

        Ack = "ACK " + str(i)
        print(f"{Ack} is received.")
        message = s.recv(1024)
        message = message.decode()

        s.send(Ack.encode())

    a = a + message

    i = i + 1

print("The message received is :", a)

```

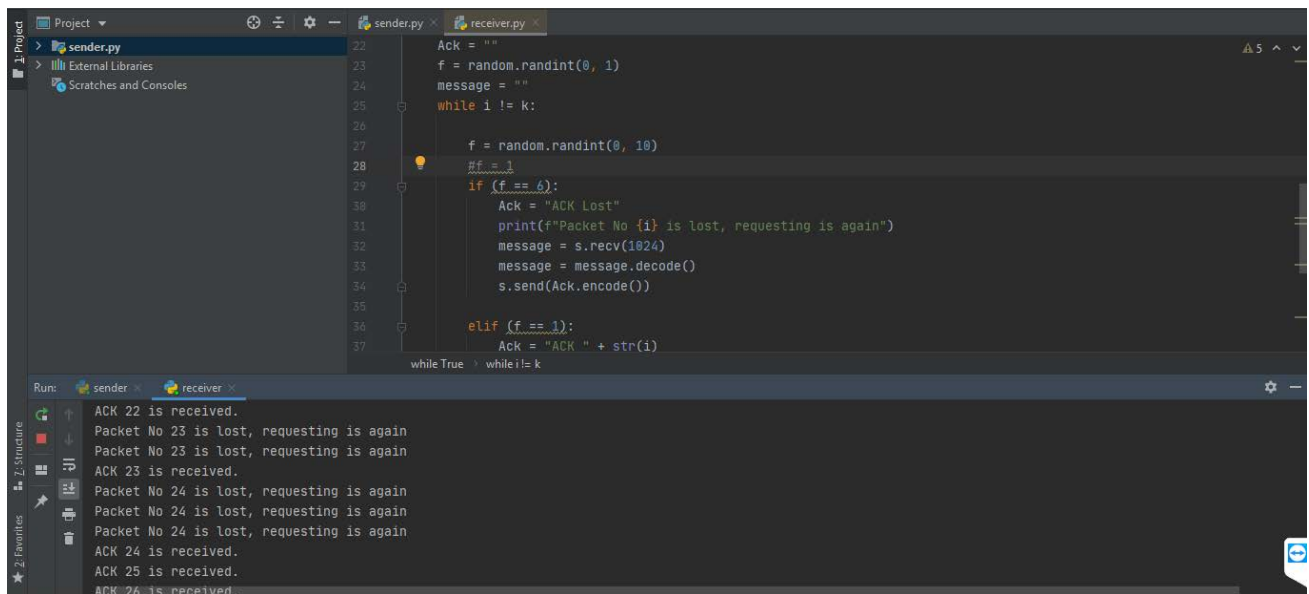
The Run console shows the following output:

```

"C:\Users\Omar D\AppData\Local\Programs\Python\Python38-32\python.exe" "E:/Faculty/Senior - 1/First Term/Computer Networks/Assignments/Go-back-N-main/receiv
Connected...

ACK 0 is received.
ACK 1 is received.
Packet No 2 is lost, requesting it again
Packet No 2 is lost, requesting it again
Packet No 2 is lost, requesting it again
Packet No 2 is lost, requesting it again
Packet No 2 is lost, requesting it again

```

```
22 Ack = ""
23 f = random.randint(0, 1)
24 message = ""
25 while i != k:
26
27     f = random.randint(0, 10)
28     #f = 1
29     if (f == 6):
30         Ack = "ACK Lost"
31         print(f"Packet No {i} is lost, requesting is again")
32         message = s.recv(1024)
33         message = message.decode()
34         s.send(Ack.encode())
35
36     elif (f == 1):
37         Ack = "ACK " + str(i)
```

while True > while i != k

Run: sender receiver

ACK 22 is received.
Packet No 23 is lost, requesting is again
Packet No 23 is lost, requesting is again
ACK 23 is received.
Packet No 24 is lost, requesting is again
Packet No 24 is lost, requesting is again
Packet No 24 is lost, requesting is again
ACK 24 is received.
ACK 25 is received.
ACK 26 is received.

Figure 4:Output When Some Packets are lost

Attached other files:

- Scanner.py (used it at first to test getting input through other file as part of distributing the code but found a way to implement adding input in the sender.py file)
- newsender.py , newreceiver.py (used them at first to test sender , receiver , used as a prototype to sender.py & receiver.py)