PA3 Report

DESIGN:

Entity A

My entity A method utilizes the go back and sending window by creating instance variables such as a sequence number limit, window size, sequence number, a building sequence number, a Queue, and a base number. The output method properly computes each packet's correct sequence number by using a modulus operator to wrap the sequence number, every time it increases after consecutive calls to the output method. The packet that is created is appended to the queue and transmitted. The trade-off of having a queue includes being able to properly perform the Go-Back-N protocol instead of having to pop packets and forget them.

The transmit method continuously loops over the packets in the queue, and sends the packets that fall into the window while increasing the sequence number and starting a timer to calculate the time out based on the factor of the window size. The trade-off of having a continuous loop to transmit packets initially was more optimized because we would not have to make a consecutive calls to send packets over to layer three and instead, we rely on as long as the sending window is capable of dispatching packets it will, and the acknowledgments will be handled in the input.

The input method continuously loops over all the packets in the sending window, which will then compare them to the packet that was received in the parameter of the input. If the packet stored in our queue in the index of our loop matches, the checkSum and the acknowledgment number of the packet that we received then we increase the base and stop the

timer if the base is equal to the sequence number otherwise we just reset the timer and transmit regardless.

The timer interrupt will simply just restart a new timer and loop over the sending window from the base plus the window size and dispatch all packets again to layer three which will arrive at entity B. This implementation was preferred over re-transmitting because it more optimally sends the correct packets in the same order that we created them.

Entity B

Entity B's design is much more simple in contrast to entity B's. The constructor will simply create a sequence number limit and a default packet and an expected sequence number of zero. Upon the input of a packet at Entity B, we will check to ensure the checksum and acknowledgment number is correct from what we expect and if so, send it off to layer five while acknowledging the same packet right back to entity since we know the packet, we will save it in the instance variable and increase the sequence number that also wraps using the modulus operator. if the packet is not good, then we send back through the last saved packet, which will always retransmit what entity previously dispatched. This was a change from what I originally implemented in programming assignment to where instead of storing the last packet I would always compute what the previous and current packet should be, but overall saving the packet in an instance variable was a much more optimal solution.

TESTING:

The testing of this project was primarily done manually and aided by the Gradescope auto grader in which I was able to properly seed and create different situations of corruption and loss, which the next section of this report will illustrate. Testing also included running through a lot of the debugger with breakpoints to ensure that packets were delivered on each side properly and with the correct sequence and acknowledgment number before any corruption was tested. When corruption was started, I implemented many print statements to illustrate what and where the packets would be lost or corrupted on which side it should be handled.

Loss 5%

python rdtsim.py -n 1000 -d 7 -z 8 -l 0.05 -c 0.0 -s 57575757578 -v 2

```
PS C:\Users\adaml\OneDrive\Documents\SPRING2024\NETWORKS\GoBackNProtocol> python rdtsim.py -n 1000 -d 7 -z 8 -l 0.05 -c 0.0 -s 57575757578
SIMULATION CONFIGURATION
-------------------------------------
(-n) # layer5 msgs to be provided:      1000
(-d) avg layer5 msg interarrival time:  7.0
(-z) transport protocol seqnum limit:   8
(-l) layer3 packet loss prob:           0.05
(-c) layer3 packet corruption prob:     0.0
(-s) simulation random seed:            57575757578
-------------------------------------

SIMULATION SUMMARY
------------------------------
# layer5 msgs provided to A:      1000
# elapsed time units:             6988.95888133036

# layer3 packets sent by A:       1174
# layer3 packets sent by B:       1112
# layer3 packets lost:            121
# layer3 packets corrupted:       0
# layer5 msgs delivered by A:     0
# layer5 msgs delivered by B:     952
# layer5 msgs by B/elapsed time:  0.13621485204943504
```

Continue to the next page

Loss 10%

python rdtsim.py -n 1000 -d 9 -z 5 -l 0.1 -c 0.0 -s 58858588855 -v 2

```
PS C:\Users\adaml\OneDrive\Documents\SPRING2024\NETWORKS\GoBackNProtocol> python rdtsim.py -n 1000 -d 9 -z 5 -l 0.1 -c 0.0 -s 58858588855
SIMULATION CONFIGURATION
-------------------------------------
(-n) # layer5 msgs to be provided:      1000
(-d) avg layer5 msg interarrival time:  9.0
(-z) transport protocol seqnum limit:   5
(-l) layer3 packet loss prob:           0.1
(-c) layer3 packet corruption prob:     0.0
(-s) simulation random seed:            58858588855
-------------------------------------

SIMULATION SUMMARY
------------------------------
# layer5 msgs provided to A:      1000
# elapsed time units:             9175.467267451198

# layer3 packets sent by A:       1227
# layer3 packets sent by B:       1120
# layer3 packets lost:            220
# layer3 packets corrupted:       0
# layer5 msgs delivered by A:     0
# layer5 msgs delivered by B:     977
# layer5 msgs by B/elapsed time:  0.10647959079596775
------------------------------
```

Corruption 5%

python rdtsim.py -n 1000 -d 7 -z 8 -l 0.0 -c 0.05 -s 129019191911 -v 2

```
PS C:\Users\adaml\OneDrive\Documents\SPRING2024\NETWORKS\GoBackNProtocol> python rdtsim.py -n 1000 -d 7 -z 8 -l 0.0 -c 0.05 -s 129019191911
SIMULATION CONFIGURATION
-------------------------------------
(-n) # layer5 msgs to be provided:      1000
(-d) avg layer5 msg interarrival time:  7.0
(-z) transport protocol seqnum limit:   8
(-l) layer3 packet loss prob:           0.0
(-c) layer3 packet corruption prob:     0.05
(-s) simulation random seed:            129019191911
-------------------------------------

SIMULATION SUMMARY
------------------------------
# layer5 msgs provided to A:      1000
# elapsed time units:             7112.190907472734

# layer3 packets sent by A:       1178
# layer3 packets sent by B:       1175
# layer3 packets lost:            0
# layer3 packets corrupted:       124
# layer5 msgs delivered by A:     0
# layer5 msgs delivered by B:     966
# layer5 msgs by B/elapsed time:  0.13582312575229524
------------------------------
```

Continue to the next page

Corruption 10%

python rdtsim.py -n 1000 -d 9 -z 8 -l 0.0 -c 0.1 -s 2626262626262 -v 2

```
PS C:\Users\adaml\OneDrive\Documents\SPRING2024\NETWORKS\GoBackNProtocol> python rdtsim.py -n 1000 -d 9 -z 8 -l 0.0 -c 0.1 -s 2626262626262
SIMULATION CONFIGURATION
-------------------------------------
(-n) # layer5 msgs to be provided:      1000
(-d) avg layer5 msg interarrival time:  9.0
(-z) transport protocol seqnum limit:   8
(-l) layer3 packet loss prob:           0.0
(-c) layer3 packet corruption prob:     0.1
(-s) simulation random seed:            2626262626262
-------------------------------------

SIMULATION SUMMARY
-------------------------------
# layer5 msgs provided to A:      1000
# elapsed time units:             8949.620908334855

# layer3 packets sent by A:       1387
# layer3 packets sent by B:       1384
# layer3 packets lost:            0
# layer3 packets corrupted:       280
# layer5 msgs delivered by A:     0
# layer5 msgs delivered by B:     946
# layer5 msgs by B/elapsed time:  0.10570280123474085
```

Loss 10% & Corruption 10%

python rdtsim.py -n 1000 -d 14 -z 8 -l 0.1 -c 0.1 -s 595959595959 -v 2

```
PS C:\Users\adaml\OneDrive\Documents\SPRING2024\NETWORKS\GoBackNProtocol> python rdtsim.py -n 1000 -d 14 -z 8 -l 0.1 -c 0.1 -s 595959595959
SIMULATION CONFIGURATION
-------------------------------------
(-n) # layer5 msgs to be provided:      1000
(-d) avg layer5 msg interarrival time:  14.0
(-z) transport protocol seqnum limit:   8
(-l) layer3 packet loss prob:           0.1
(-c) layer3 packet corruption prob:     0.1
(-s) simulation random seed:            595959595959
-------------------------------------

SIMULATION SUMMARY
-------------------------------
# layer5 msgs provided to A:      1000
# elapsed time units:             13969.217573281685

# layer3 packets sent by A:       1735
# layer3 packets sent by B:       1566
# layer3 packets lost:            317
# layer3 packets corrupted:       287
# layer5 msgs delivered by A:     0
# layer5 msgs delivered by B:     974
# layer5 msgs by B/elapsed time:  0.06972473546857251
-------------------------------
```

All instances of loss and corruption were handled as shown in the proof above of the simulation output which will go to aid the next section of this report when filing the output. No cases which the simulation failed had occurred, therefore I am confident that my implementation is correct.

OUTPUT:

The following print statements will illustrate the output for the conditions assigned.

Entity A

```
Adam Liu *
def transmit(self):
    if TRACE > 0:
        print("Transmitting packets with base: "+ str(self.base) + ", With window size: " + str(self.window_size))
    # Continuously loop over the window and send out packets
    while (self.next_seqnum < self.base + self.window_size) and (self.next_seqnum < len(self.q)):
        to_layer3(self, self.q[self.next_seqnum])

        # Start the timer and increase the sequence number
        if self.base == self.next_seqnum:
            start_timer(self, 10 + 4 * self.window_size)
        self.next_seqnum += 1

Adam Liu *
def input(self, packet):

    # Loop over all packets in the window and compare to the received packet
    if TRACE > 0:
        print("Packet received, checking window from base: "+ str(self.base) + ", With window size: " + str(self.window_size))
    for i in range(self.base, self.base + self.window_size):
        if i >= len(self.q):
            break

        # If the packet is valid, increase the base and stop or reset the timer
        if self.q[i].checksum == packet.checksum and self.q[i].seqnum == packet.acknum:
            if TRACE > 0:
                print("Packet with sequence number: " + str(packet.seqnum) + " acknowledged at Entity A")
            self.base = i + 1
            if self.base == self.next_seqnum:
                stop_timer(self)
            else:
                stop_timer(self)
                start_timer(self, 10 + 4 * self.window_size)

    # Transmit the next window of packets
    self.transmit()
```

Entity B

```
Adam Liu *
def input(self, packet):

    # Calculate the expected checksum and compare the seqnum of the input
    checksum = packet.seqnum + packet.acknum + sum(packet.payload)
    if packet.seqnum == self.expected_seqnum and packet.checksum == checksum:
        if TRACE > 0:
            print("Packet with sequence number: " + str(packet.seqnum) + " acknowledged at Entity B")
        # Send the packet off and acknowledge to entity A
        to_layer5(self, Msg(packet.payload))
        to_layer3(self, packet)

        # Save the most recent valid packet
        self.last_pkt = packet
        self.expected_seqnum = (self.expected_seqnum + 1) % self.seqnum_limit

    # Otherwise send back the most recent valid acknowledged packet
    else:
        if TRACE > 0:
            print("Packet corrupted at Entity B, sending back acknowledgement: " + str(self.last_pkt.seqnum))
        to_layer3(self, self.last_pkt)
```

Loss 5% & Corruption 5%

python rdtsim.py -n 1000 -d 14 -z 8 -l 0.05 -c 0.05 -s 595959595959 -v 2

Below shows the log of a few packets being delivered and its entire route annotated.

```
Packet with sequence number: 6 acknowledged at Entity B
Packet received, checking window from base: 981, With window size: 4
Packet with sequence number: 5 acknowledged at Entity A
Transmitting packets with base: 982, With window size: 4
Packet received, checking window from base: 982, With window size: 4
Packet with sequence number: 6 acknowledged at Entity A
Transmitting packets with base: 983, With window size: 4
Packet with sequence number: 7 acknowledged at Entity B
Packet received, checking window from base: 983, With window size: 4
Packet with sequence number: 7 acknowledged at Entity A
Transmitting packets with base: 984, With window size: 4
Packet corrupted at Entity B, sending back acknowledgement: 7
Packet received, checking window from base: 984, With window size: 4
Transmitting packets with base: 984, With window size: 4
Packet corrupted at Entity B, sending back acknowledgement: 7
Transmitting packets with base: 984, With window size: 4
Packet received, checking window from base: 984, With window size: 4
Transmitting packets with base: 984, With window size: 4
Packet corrupted at Entity B, sending back acknowledgement: 7
Packet received, checking window from base: 984, With window size: 4
Transmitting packets with base: 984, With window size: 4
Transmitting packets with base: 984, With window size: 4
Timeout occurred, resending window from base: 984, With window size: 4
Packet corrupted at Entity B, sending back acknowledgement: 7
Packet with sequence number: 0 acknowledged at Entity B
Packet with sequence number: 1 acknowledged at Entity B
Packet received, checking window from base: 984, With window size: 4
Transmitting packets with base: 984, With window size: 4
Packet with sequence number: 2 acknowledged at Entity B
Packet with sequence number: 3 acknowledged at Entity B
Packet received, checking window from base: 984, With window size: 4
Packet with sequence number: 0 acknowledged at Entity A
Transmitting packets with base: 985, With window size: 4
Transmitting packets with base: 985, With window size: 4
Packet with sequence number: 4 acknowledged at Entity B
```

This Image illustrates how 20 messages were successfully transferred from sender to receiver (The sender received ACKs for these messages), a loss probability of 0.05, a corruption probability of 0.05, and a trace level of 2. As expected, the acknowledgement numbers wrap in a sequence as specified by Go-Back-N and my implementation is full and complete.

```
===== SIMULATION BEGINS
Packet with sequence number: 0 acknowledged at Entity B
Packet with sequence number: 0 acknowledged at Entity A
Packet with sequence number: 1 acknowledged at Entity B
          TO_LAYER3: packet being lost
Packet with sequence number: 2 acknowledged at Entity B
Packet with sequence number: 2 acknowledged at Entity A
Packet with sequence number: 3 acknowledged at Entity B
Packet with sequence number: 3 acknowledged at Entity A
Packet with sequence number: 4 acknowledged at Entity B
          TO_LAYER3: packet being corrupted
Packet with sequence number: 4 acknowledged at Entity A
          TO_LAYER3: packet being lost
Packet with sequence number: 5 acknowledged at Entity B
Packet with sequence number: 5 acknowledged at Entity A
Packet with sequence number: 6 acknowledged at Entity B
Packet with sequence number: 6 acknowledged at Entity A
Packet with sequence number: 7 acknowledged at Entity B
Packet with sequence number: 7 acknowledged at Entity A
Packet with sequence number: 0 acknowledged at Entity B
Packet with sequence number: 0 acknowledged at Entity A
Packet with sequence number: 1 acknowledged at Entity B
Packet with sequence number: 1 acknowledged at Entity A
Packet with sequence number: 2 acknowledged at Entity B
          TO_LAYER3: packet being lost
Packet with sequence number: 3 acknowledged at Entity B
Packet with sequence number: 3 acknowledged at Entity A
Packet with sequence number: 4 acknowledged at Entity B
          TO_LAYER3: packet being lost
Packet with sequence number: 5 acknowledged at Entity B
Packet with sequence number: 5 acknowledged at Entity A
Packet with sequence number: 6 acknowledged at Entity B
Packet with sequence number: 6 acknowledged at Entity A
Packet with sequence number: 7 acknowledged at Entity B
Packet with sequence number: 7 acknowledged at Entity A
Packet with sequence number: 0 acknowledged at Entity B
          TO_LAYER3: packet being lost
Packet with sequence number: 1 acknowledged at Entity B
Packet with sequence number: 1 acknowledged at Entity A
Packet with sequence number: 2 acknowledged at Entity B
Packet with sequence number: 2 acknowledged at Entity A
Packet with sequence number: 3 acknowledged at Entity B
Packet with sequence number: 4 acknowledged at Entity B
Packet with sequence number: 5 acknowledged at Entity B
Packet with sequence number: 3 acknowledged at Entity A
Packet with sequence number: 4 acknowledged at Entity A
```

```
===== SIMULATION ENDS

SIMULATION SUMMARY
-------------------------------
# layer5 msgs provided to A:      1000
# elapsed time units:             14093.98076844215

# layer3 packets sent by A:       1385
# layer3 packets sent by B:       1321
# layer3 packets lost:            143
# layer3 packets corrupted:       146
# layer5 msgs delivered by A:     0
# layer5 msgs delivered by B:     995
# layer5 msgs by B/elapsed time:  0.07059751367249668
-------------------------------
PS C:\Users\adaml\OneDrive\Documents\SPRING2024\NETWORKS\GoBackNProtocol>
```