Section A

Overview:

The task here is to analyze the TCP packets by building a tool similar to Wireshark.

Code:

1. Class **Packet** is used to read the packet bytes and ascertain the packet attributes. Function **parse** is used to initialize the packet attributes.
2. Class **Connection** is used to ascertain the flow of the packets between the source and destination ports.
3. Function **ParseConnections** is used to assign ports and track the flow of each packet.
4. Function **getTransactionDetails** is used to find transaction details for every flow.
5. Function **getValues** is used to fetch sequence number, acknowledgement number and the receive window size for the first two transactions after the TCP connection is set up. We find transactions for both sides - sender to receiver and vice-versa.
6. Function **findThroughput** is used to calculate the empirical throughput by dividing the total payload size by time. It returns a result in megabytes per second.
7. Function **computeLossRate** is used to calculate the packet loss by counting the number of times a packet is sent after the initial transmission. Loss rate is calculated by dividing packet loss by total number of packets.
8. Function **calculateRTT** is used to measure the round trip time. It also calculates the theoretical output.

Output:

1. **Number of TCP Flows:**

```
1. Number of TCP flows initiated by the sender: 3
80:43498:11106
80:43500:11834
80:43502:1185
```

2. **For each TCP Flow:**
    a. **Transaction Details**

```
2. Transaction Details (2 per TCP connection):

> Connecting 80 to 43498 ----
Sender to Receiver:

Packet 1: SEQ -  705669103  ACK -  1921750144  WND -  3
Packet 2: SEQ -  705669127 ACK -  1921750144  WND -  3
```

```
Receiver to Sender:
Packet  1 :  SEQ -  1921750144 ACK -  705669127 WND -  3
Packet  2 :  SEQ -  1921750144 ACK -  705670575 WND -  3

> Connecting 80 to 43500 ----
Sender to Receiver:

Packet 1: SEQ -  3636173852  ACK -  2335809728  WND -  3
Packet 2: SEQ -  3636173876 ACK -  2335809728  WND -  3

Receiver to Sender:
Packet  1 :  SEQ -  2335809728 ACK -  3636173876 WND -  3
Packet  2 :  SEQ -  2335809728 ACK -  3636175324 WND -  3

> Connecting 80 to 43502 ----
Sender to Receiver:

Packet 1: SEQ -  2558634630  ACK -  3429921723  WND -  3
Packet 2: SEQ -  2558634654 ACK -  3429921723  WND -  3

Receiver to Sender:
Packet  1 :  SEQ -  3429921723 ACK -  2558634654 WND -  3
Packet  2 :  SEQ -  3429921723 ACK -  2558636102 WND -  3
```

b. Throughput

```
3. Throughput : 5.251391112912558  MBps

3. Throughput : 1.285420726825806  MBps

3. Throughput : 1.4815063848257195  MBps
```

c. Loss rate

```
4. Packets Lost:  3
Loss Rate:  0.0004299842339114232

4. Packets Lost:  94
Loss Rate:  0.013299377475947935

4. Packets Lost:  0
Loss Rate:  0.0
```

d. Average RTT and Theoretical throughput

```
5. Round Trip Time:  0.07300400733947754  seconds
Theoretical Throughput:  1.1812056601104137  MBps

5. Round Trip Time:  0.07270503044128418  seconds
Theoretical Throughput:  0.21326440153775036  MBps

5. Round Trip Time:  0.07350778579711914  seconds
Theoretical Throughput Error: Infinity (division by zero)
```

Explanation:

1. Number of TCP Flows:

We look at unique sets of source port and destination port pairs to calculate the number of TCP flows.

2. For each TCP flow:

a. Transaction details

We look at each flow and analyze the first two packets. However, we skip packet 1 which is the SYN, packet 2 which is the SYN-ACK and packet 3 which is the ACK. So we consider 4th packet (index = 3) and 5th packet (index = 4) as the first two transactions from the sender to the receiver. When looking at the other way, we consider packets with SEQ number equal to the ACK of the packets sent from sender to receiver.

For each of the packet above, we print the sequence number, acknowledgement number and the receiver window size, which is multiplied by 2 raised to scaling factor, to compute the receiver's buffer size.

b. Throughput

We calculate throughput by dividing the total size of all packets (total payload size) by the time difference between the first and the last packet. We ignore the ACKs from the sender since we calculate throughput at the receiver.

c. Loss Rate

We calculate the loss rate by counting the number of times a particular packet is transmitted, identified by its source IP, destination IP and the sequence number. We subtract 1 to consider only the number of times the packet was retransmitted. This gives us packet loss and dividing packet loss by total number of packets (considering initial transmissions) gives us loss rate.

d. Average RTT

In this part, we calculate the round trip times by using a weighted average of the past RTTs:

$RTT = \alpha * RTT_{new} + (1 - \alpha) * RTT_{old}$

We take the value of alpha as 0.125.

We calculate the initial RTT from the SYN and SYN-ACK packets. We put all packets from sender to receiver in one dictionary and all packets from receiver to sender in another dictionary. Then we map the ACK to the corresponding SEQ, and subtract an empirical value of payload (1448). We find RTT_{new} by calculating the difference between the ACK and SEQ dictionary packets.

In line with Karn's algorithm, we ignore the retransmitted samples.

e. Theoretical Throughput

We calculate theoretical throughput by using the formula: $\frac{MSS\ * \sqrt{3}}{\sqrt{2} * \sqrt{loss\ rate} * RTT}$

Throughput (MBps)	Theoretical	Empirical
Flow 1	1.181 MBps	5.251 MBps
Flow 2	0.213 MBps	1.285 MBps
Flow 3	Infinity (division by zero)	1.481 MBps

We see that theoretical throughput turns out to be lower than the empirical throughputs. This is the opposite of what we expected as theoretical values do not take into account the unprecedented network errors. The only possible explanation is that the formula seems to be missing some updates.