Part A:

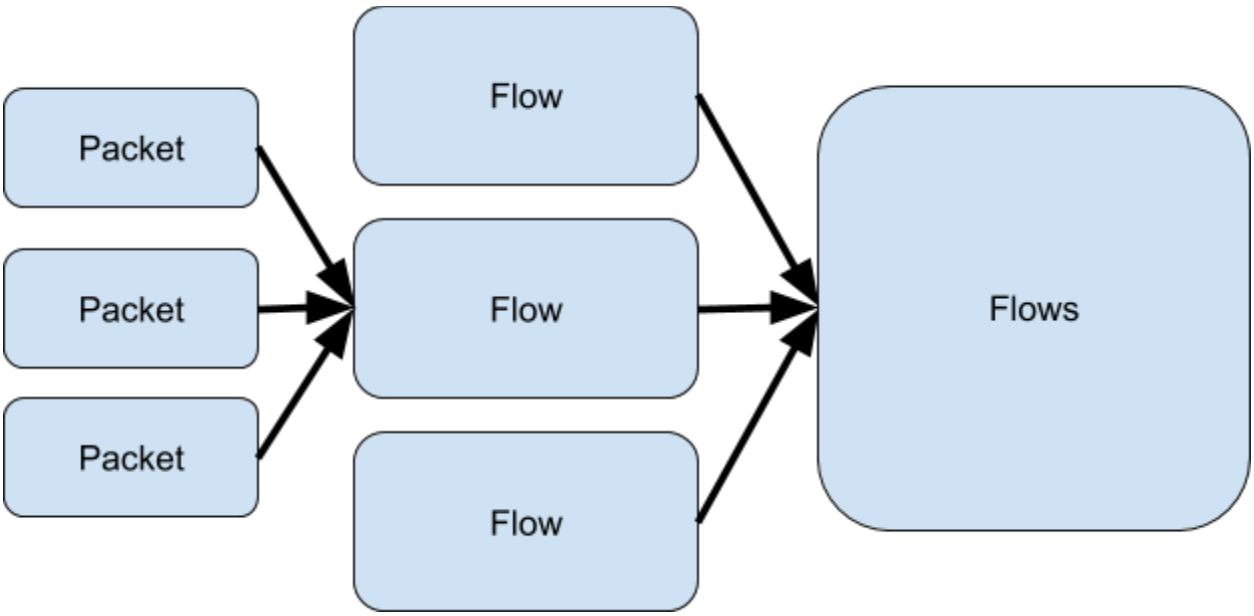
High Level View of the Program:

The program is written in python 3 and can be run through the command: python analysis_pcap_tcp.py . Running this will ask which file you would like to use and prompt you for what you would like to inspect.

Each packet is structured as a dictionary with attributes of “TIME”, “SIZE”, “SOURCE” (source port), “DESTINATION” (destination port), “SEQUENCE” (sequence number), “ACK” (acknowledgement number), “HEAD LENGTH”, “FLAG”, “FLAG SEP” (flags separated, in a sequence of 1’s and 0’s marking presence (1) or absence (0) of a flag), “WINDOW” (window size), “CHECKSUM”, “URGENTPOINTER”, “PAYLOAD” and “SCALE”.

A single flow is represented as a list of packets that composes it. This flow contains only interactions between two specific ports. The global list: flows is a list composed of these flows. In list format, this can be depicted as:

flows = [[flow1packet1, flow1packet2,...],[flow2packet1,flow2packet2,...],..], or as an image, it can be depicted as such:



Answers to Questions:

1. The number of TCP flows initiated from the sender is 3. They are from the ports: 43498, 43500, and 43502.
2.
a.

Source Port: 43498:

Transaction 1:

Sender:

Sequence Number: 705669103
Ack Number: 1921750144
Window Size: 3

Receiver:

Sequence Number: 1921750144
Ack Number: 705669127
Window Size: 3

Transaction 2:

Sender:

Sequence Number: 705669127
Ack Number: 1921750144
Window Size: 3

Receiver:

Sequence Number: 1921750144
Ack Number: 705670575
Window Size: 3

Source Port 43500:

Transaction 1:

Sender:

Sequence Number: 3636173852
Ack Number: 2335809728
Window Size: 3

Receiver:

Sequence Number: 2335809728
Ack Number: 3636173876
Window Size: 3

Transaction 2:

Sender:

Sequence Number: 3636173876
Ack Number: 2335809728
Window Size: 3

Receiver:
Sequence Number: 2335809728
Ack Number: 3636175324
Window Size: 3

Source Port 43502:

Transaction 1:
Sender:
Sequence Number: 2558634630
Ack Number: 3429921723
Window Size: 3

Receiver:
Sequence Number: 3429921723
Ack Number: 2558634654
Window Size: 3

Transaction 2:
Sender:
Sequence Number: 2558634654
Ack Number: 3429921723
Window Size: 3

Receiver:
Sequence Number: 3429921723
Ack Number: 2558636102
Window Size: 3

For each of these the sequence number represents the byte that we start off at, the ack number represents the next byte that we expect to receive, and the window size represents that we can receive the number given times 16384 bytes of data. For example, in the first transaction, we say that we start off at the 705669103th byte, and we expect the next byte to be at the 1921750144th byte. The window size would be 3*16384 which is equal to 49152 bytes. For all transactions within this example, the window size is equal to 49152 bytes.

b.

Source Port: 43498
Throughput: 43.096 Mbps

Source Port: 43500
Throughput: 10.586 Mbps

Source Port: 43502
Throughput: 12.177 Mbps

Throughput in this example was calculated by summing up the total number of bytes divided by the total time. The number of bytes included the payload as well as the TCP header, Link header, and IP header.

c.

Source Port: 43498
Total Number of Packets: 11106
Number of Lost Packets: 3
Rate of Loss: 0.00027012

Source Port: 43500
Total Number of Packets: 11834
Number of Lost Packets: 94
Rate of Loss: 0.0079432

Source Port: 43502
Total Number of Packets: 1185
Number of Lost Packets: 0
Rate of Loss: 0.0

In order to calculate loss rate, I counted the number of packets lost and divided that by the total number of packets. The number of packets lost were taken to be the number of those that were retransmitted. When packets are retransmitted, it is done because it is assumed that the packet has been lost somehow.

d.

Source Port 43498
RTT: 0.073543 seconds
Experimental Throughput: 43.096 Mbps
Theoretical Throughput: 11.835 Mbps

Source Port 43500
RTT: 0.088603 seconds
Experimental Throughput: 10.586 Mbps
Theoretical Throughput: 1.8115 Mbps

Source Port 43502
RTT: 0.073274 seconds
Experimental Throughput: 12.177 Mbps
Theoretical Throughput: infinity

In this example, to estimate RTT, I took the time that the ack was received minus the time that the packet was sent. I took the average of this over all packets (which successfully got acknowledged, and did not get retransmitted). I divided this by the number of packets this occurred with. I took the theoretical throughput to be equal to the number of packets lost over the total number of packets divided by the square root of rate loss times RTT ($(packet\ loss \div total\ packets) \div (\sqrt{rate\ loss} \times RTT)$).

The formula $(1.22 \times MSS) \div (\sqrt{L} \times RTT)$ can be used in order to solve for the loss rate given the experimental throughput. Using source port 43498, we can see that the experimental throughput is about 43 Mbps. Plugging in this number, this results in a loss rate of about 2*10-5, which in turn means that one in every 50000 should be lost. However, since we only have about 1/5th of this (11106 segments), this is impossible. In this scenario, using this formula, empirical throughput would always be greater than theoretical. It is possible, given recent advances, that this has become an outdated formula.