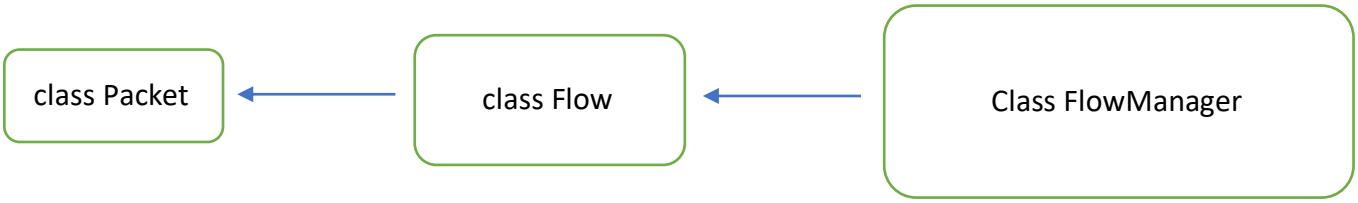
Table of Contents

1 High Level View 1
2 Answers 2
2.1 PART A(a) 2
2.2 PART A(b) 3
2.3 PART A(c) 3
2.4 PART A(d) 4

1 High Level View

There are three classes: Packet, Flow, and FlowManager



1. Class Packet encapsulates all the information in a packet. The most important method of this packet is **parsing** the **raw bytes** into **human readable TCP fields**.
2. Class Flow encapsulates a list of all packets in a flow. A flow is identified by unique **port pair (source, dest)**. This class contains important methods including **compute_throughput()**, **compute_loss_rate()**, **estimateRTT()**, **compute_dta_timeout()**
3. Class FlowManager encapsulates all the flows in a pcap file. It manages everything.

The high level view of most algorithms is to put the packet in dictionaries { seq → packet} or { ack →packet} and then do the corresponding mapping.

How to run the program:

```
caitaos-mbp:homework-2 caitao$ python analysis_pcap_tcp.py
```

The program is written in Python3

2 Answers

```
init a new flow 100
init a new flow 101
init a new flow 102

There are 3 TCP flows initiated from the sender
```

2.1 PART A(a)

```
***Flow 100***

Transaction 1:
Sender:  sequence # = 705669103  acknowledge # = 1921750144  receive window = 3
I just a sent some data to you, starting at the (705669103)th byte.
The next btye I anticipate to receive from you starts at the (1921750144)th byte.
My receive window size is 3 times 16384.

Receiver: sequence # = 1921750144  acknowledge # = 705669127  receive window = 3
I just a sent some data to you, starting at the (1921750144)th byte.
The next btye I anticipate to receive from you starts at the (705669127)th byte.
My receive window size is 3 times 16384.

Transaction 2:
Sender:  sequence # = 705669127  acknowledge # = 1921750144  receive window = 3
I just a sent some data to you, starting at the (705669127)th byte.
The next btye I anticipate to receive from you starts at the (1921750144)th byte.
My receive window size is 3 times 16384.

Receiver: sequence # = 1921750144  acknowledge # = 705670575  receive window = 3
I just a sent some data to you, starting at the (1921750144)th byte.
The next btye I anticipate to receive from you starts at the (705670575)th byte.
My receive window size is 3 times 16384.




***Flow 101***

Transaction 1:
Sender:  sequence # = 3636173852  acknowledge # = 2335809728  receive window = 3
I just a sent some data to you, starting at the (3636173852)th byte.
The next btye I anticipate to receive from you starts at the (2335809728)th byte.
My receive window size is 3 times 16384.

Receiver: sequence # = 2335809728  acknowledge # = 3636173876  receive window = 3
I just a sent some data to you, starting at the (2335809728)th byte.
The next btye I anticipate to receive from you starts at the (3636173876)th byte.
My receive window size is 3 times 16384.

Transaction 2:
Sender:  sequence # = 3636173876  acknowledge # = 2335809728  receive window = 3
```

```
I just a sent some data to you, starting at the (3636173876)th byte.
The next btye I anticipate to receive from you starts at the (2335809728)th byte.
My receive window size is 3 times 16384.

Receiver: sequence # = 2335809728  acknowledge # = 3636175324  receive window = 3
I just a sent some data to you, starting at the (2335809728)th byte.
The next btye I anticipate to receive from you starts at the (3636175324)th byte.
My receive window size is 3 times 16384.


***Flow 102***

Transaction 1:
Sender:  sequence # = 2558634630  acknowledge # = 3429921723  receive window = 3
I just a sent some data to you, starting at the (2558634630)th byte.
The next btye I anticipate to receive from you starts at the (3429921723)th byte.
My receive window size is 3 times 16384.

Receiver: sequence # = 3429921723  acknowledge # = 2558634654  receive window = 3
I just a sent some data to you, starting at the (3429921723)th byte.
The next btye I anticipate to receive from you starts at the (2558634654)th byte.
My receive window size is 3 times 16384.

Transaction 2:
Sender:  sequence # = 2558634654  acknowledge # = 3429921723  receive window = 3
I just a sent some data to you, starting at the (2558634654)th byte.
The next btye I anticipate to receive from you starts at the (3429921723)th byte.
My receive window size is 3 times 16384.

Receiver: sequence # = 3429921723  acknowledge # = 2558636102  receive window = 3
I just a sent some data to you, starting at the (3429921723)th byte.
The next btye I anticipate to receive from you starts at the (2558636102)th byte.
My receive window size is 3 times 16384.
```

2.2 PART A(b)

To estimate throughput, I counted **data** including the payload and all three headers in every packet. Three headers include TCP header, IP header, and Link header. **Time** is the timestamp difference between the first packet and the last packet. Therefore, estimate throughput is **data**÷**time**.

```
***Flow 100***
Throughput is 43.09558 Mbps

***Flow 101***
Throughput is 10.58582 Mbps

***Flow 102***
Throughput is 12.17738 Mbps
```

2.3 PART A(c)

```
# of loss equals # of retransmission, since when we retransmit, we assume the packe
t is loss.

***Flow 100***
# of loss is 3
# of packets send is 11106
Therefore, the loss rate is 0.000270

***Flow 101***
# of loss is 94
# of packets send is 11834
Therefore, the loss rate is 0.007943

***Flow 102***
# of loss is 0
# of packets send is 1185
Therefore, the loss rate is 0.000000
```

2.4 PART A(d)

```
***Flow 100***
Estimated RTT is 0.07354 second
Theoretical throughput is 11.83498 Mbps

***Flow 101***
Estimated RTT is 0.08860 second
Theoretical throughput is 1.81151 Mbps

***Flow 102***
Estimated RTT is 0.07327 second
Theoretical throughput is infinity
```

Compare

Theoretical throughput < empirical throughput.

Explain

When using the formula $\frac{1.22 \cdot MSS}{RTT\sqrt{L}}$ to achieve the empirical 43 Mbps in the first flow, we need a loss rate at 2×10^{-5}, that is one loss event for every 50,000 segments. However, we only have 11,106 segments in the first flow. Therefore, by using the formula, there is no way to achieve the empirical throughput.

According to KS textbook, this is because the TCP congestion control has evolved over the years. And now we are in the age of high-bandwidth environments. In the high-bandwidth age, the formula above may not be accurate or even appropriate anymore.