COMP556 Assignment1 Part 2

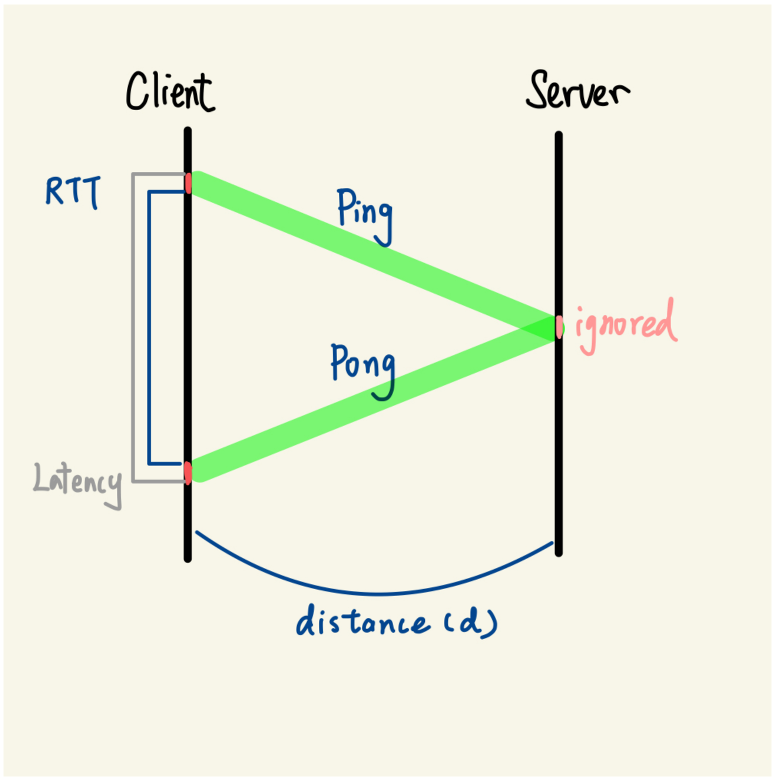
1. Concept

We aim to measure the Bandwidth Independent Delay (BID). Since RTT includes the package transmission time affected by bandwidth, i.e. Bandwidth Dependent Delay (BDD), we calculate the Bandwidth Independent Delay by subtracting the Bandwidth Dependent Delay from RTT.

$$RTT - BDD = BID$$

Then, how can we verify that the BID we calculated is correct? We will verify it by sending packets with different data sizes.When the data size increases, the BID increases and the two have a linear relationship as mentioned in the specifications, this verifies the accuracy of our BID results.

2. Experiment



- We designed to send 655 packets with data size from 18 to 65518, from the client to the server and measured the total time spent.
- We will record a timestamp before and after the **send()** function tocalculate the dependent delay.
- When Calculating RTT, we ignored the latency for the ack package since it is small.
- After testing, we would made the latencies we observed and the data sizes into pair.
- In order to visual data, we used RANSACRegressor package to deal with outliers and find the line of best fit.
- Plot implementation Code(Python):

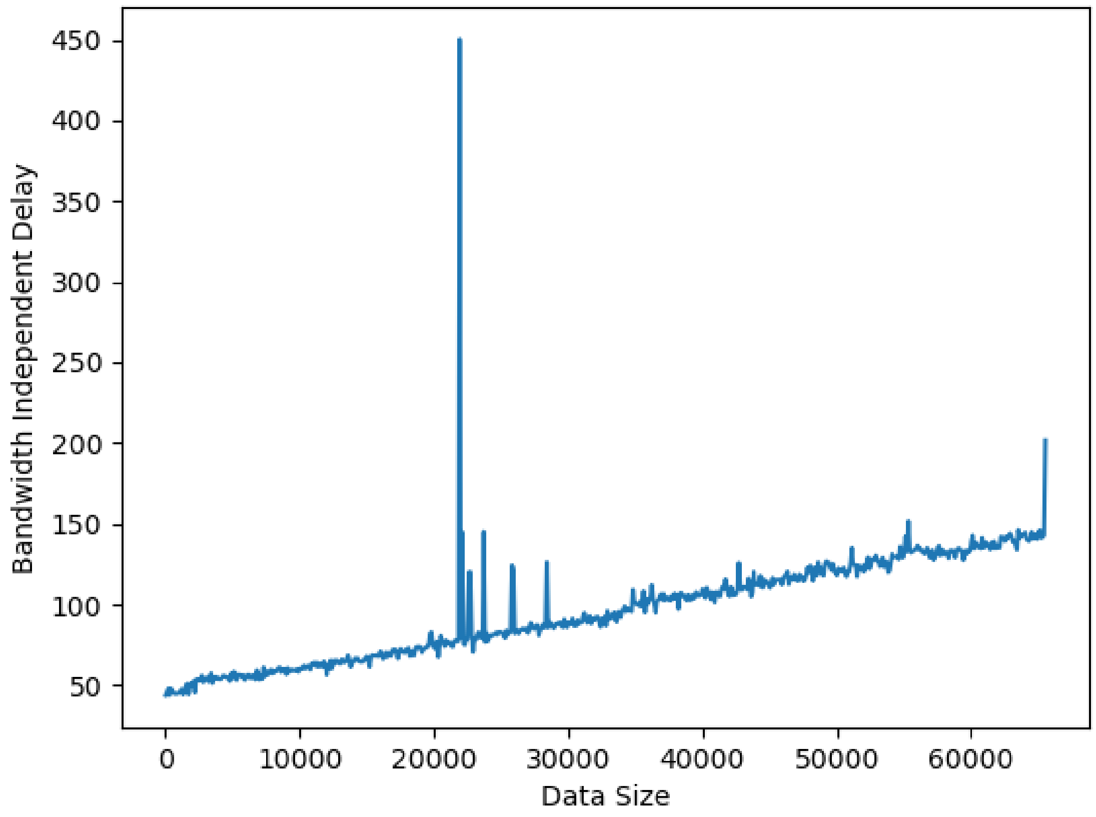
3. Result

Bandwidth Independent Delay:

- Plot implementation Code(Python):

```python
import matplotlib.pyplot as plt
import numpy as np

def calBID(rtt: float, bandwidthDelay: float) -> float:
    return rtt - 2*bandwidthDelay

if __name__ == "__main__":
    filename = "times2.txt"
    xList = []
    yList = []
    with open(filename, "r") as fp:
        lines = fp.readlines()
        for line in lines:
            dataSize, count, bandwidthDelay, rtt = line.split(", ")
            dataSize = int(dataSize)
            count = int(count)
            bandwidthDelay = float(bandwidthDelay)
            rtt = float(rtt)

            bid = calBID(rtt, bandwidthDelay)
            xList.append(dataSize)
            yList.append(bid)

    x = np.array(xList)
    y = np.array(yList)

    plt.plot(x, y)
    plt.xlabel("Data Size")
    plt.ylabel("Bandwidth Independent Delay")
    plt.show()
```

- Plot:



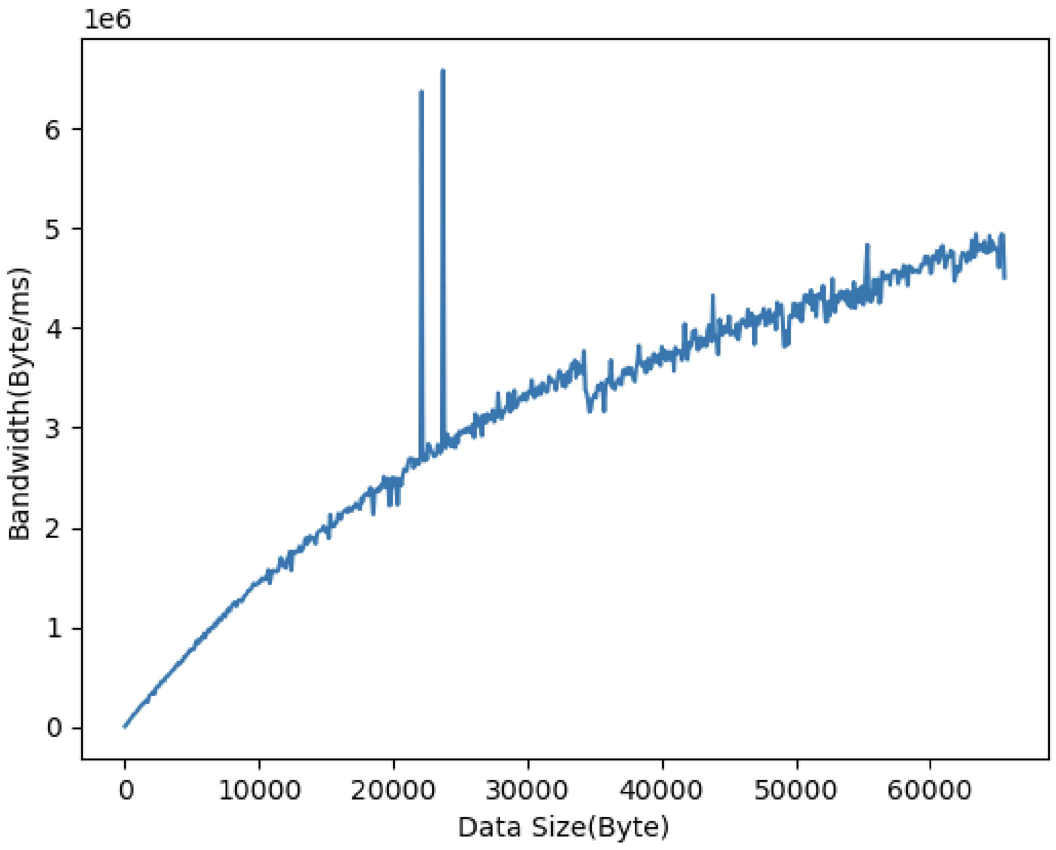
- The y_intercept of the line corresponds to Bandwidth Independent Delay and the x_intercept of the line corresponds to Data Size.
- From the trend line in the graph, when the datasize we sent become bigger, the BID would also become bigger too.
- It can be seen that the data on the x and y axes are proportional to each other, thereby proving our hypothesis to be correct.

Bandwidth:

- Plot implementation Code(Python):

```
def bandwidthFigure():
    filename = "times2.txt"
    xList = []
    yList = []
    with open(filename, "r") as fp:
        lines = fp.readlines()
        for line in lines:
            dataSize, count, bandwidthDelay, latency = line.split(", ")
            dataSize = int(dataSize)
            count = int(count)
            bandwidthDelay = float(bandwidthDelay)
            latency = float(latency)

            bid = calBID(latency, bandwidthDelay)
            xList.append(dataSize)
            yList.append(2*dataSize*10000/(latency - bid))

    x = np.array(xList)
    y = np.array(yList)

    plt.plot(x, y)
    plt.xlabel("Data Size(Byte)")
    plt.ylabel("Bandwidth(Byte/ms)")
    plt.show()
```

- Plot:


1e6
Bandwidth(Byte/ms)
0
1
2
3
4
5
6
0
10000
20000
30000
40000
50000
60000
Data Size(Byte)

4. Discussion

There are also some isssue we would like to discuss, trying to explained the outlier and error issues.

- Resource management problem:

  When calling the send function, there are overhead considerations with regards to the network and the operating system. When passing a buffer address to the send function, the address does not directly correspond to a physical address, resulting in overhead from the operating system, which is a clear issue with resource management.

- Packet transmission time overlapping problem:

  When the data size is too large, it will not only send one package, but multiple packages. This will result in inaccuracies in the calculation of independent delay, as the server will return an acknowledgment for the first received package before the client has finished sending all packages. As a result, when the data size increases, the calculation error of transmission time will also increases.