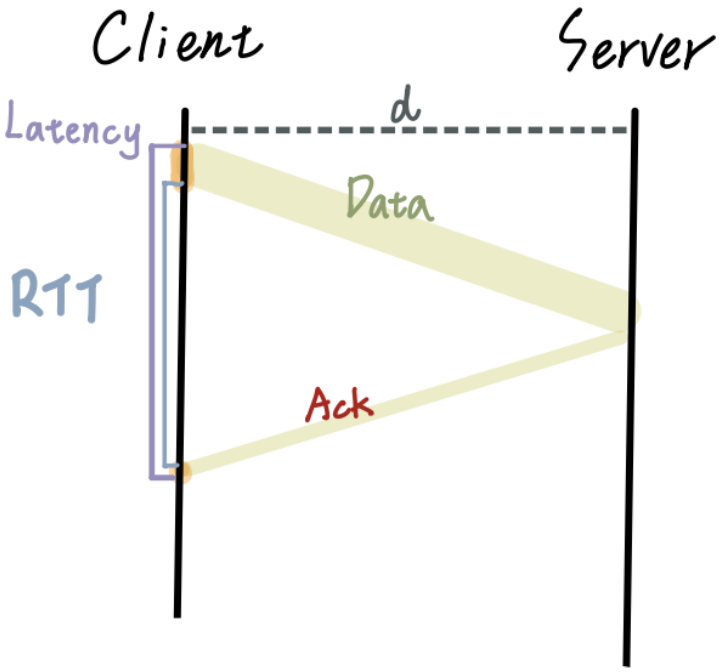
Part2

I. Definition



- Data size and bandwidth dependent delay
  → $\frac{data\_size}{bandwidth}$ ... D_d
  * Transmission time is bandwidth dependent
- Data size and bandwidth independent delay
  → $\frac{d}{propagation\_rate}$ ... D_i
  * RTT is bandwith independent

Latency = Transmission time (data) + Transmission time (ACK) + RTT

⚠ Goal: estimate RTT

II. Concept

- We have 2 data with different size: D_1, D_2

1° The client sents D_1 to the server (small enough to be ignored)

$$\text{Latency\_1} = \text{Transmission time (data)} + \text{Transmission time (ACK)} + \text{RTT}$$
$$= \frac{D_1\_size}{bandwidth} + RTT \quad \dots (1)$$

2° The client sents D_2 to the server

$$\text{Latency\_2} = \frac{D_2\_size}{bandwidth} + RTT \quad \dots (2)$$

Since we already calculated Latency_1.
Latency_2 in part 1, we could simply
compute bandwidth and RTT using (1) & (2)

- In this part, we assume RTT is not related to data size

I. $(1) - (2) = \frac{D_1\_size - D_2\_size}{bandwidth}$ ⇒ get bandwidth B

II. $RTT = (1) - \frac{D_1\_size}{B}$

III. Estimation

- We sent 600 packets with different data size from the client to the server and measured the latency. After that, we stored the data size and corresponding latency in pairs.
- Next, we use RANSACRegressor to deal with outliers and find the line of best fit.
- The y_intercept of the line corresponds to “RTT”; the slope of the line corresponds to “Bandwidth”.

IV. Experiment

- `size_arr` is used to store data size and its shape is (600, 1).
- `latency_arr` is used to store latency and its shape is (600, 1).

```
def plot_best_fit(X, y, model):
    model.fit(X, y)
    print('Bandwidth(Slope): %.3f' % model.estimator_.coef_[0])
    print('RTT(Intercept): %.3f' % model.estimator_.intercept_)

    plt.scatter(X, y)
    # plot the line of best fit
    xaxis = arange(X.min(), X.max(), 0.001)
    yaxis = model.predict(xaxis.reshape((len(xaxis), 1)))
    plt.plot(xaxis, yaxis, color='r')

    # show the plot
    plt.title(type(model).__name__)
    plt.xlabel('Data Size(byte)')
    plt.ylabel('Latency(μs)')
    plt.show()

X = np.expand_dims(np.array(size_arr), axis=1)
y = np.expand_dims(np.array(latency_arr), axis=1)
model = RANSACRegressor()

# plot the line of best fit
plot_best_fit(X, y, model)
```

V. Result

```
Bandwidth(Slope): 0.001
RTT(Intercept): 96.334
```