

شبکه‌های کامپیوتری

تمرین سری یک

شماره دانشجویی: ۹۷۱۰۳۷۷۹

نام و نام خانوادگی: آئیریا محمدی

۱. شرح کد client.py در ادامه آمده است.

محاسبه چهار خواسته سوال:

```
5
6 for _ in range(10):
7     start = time()
8     msg()
9     recv()
10    times.append(time()-start)
11
12 avg_time = sum(times)/10
13 print(f"1-1: {avg_time/n}")
14
15 # 1-2
16 start = time()
17
18 for _ in range(10):
19     msg()
20     recv()
21
22 avg_time = (time() - start)/10
23 print(f"1-2: {avg_time/n}")
24
25 # 1-3
26 start = time()
27 client.sendto(("throughput"*10).encode(), serverAddressPort)
28 recv()
29 duration = time() - start
30 print(f"1-3: {duration/(n*10)}")
31
32 # 1-4
33 start = time()
34 client.sendto(("throughput"*100).encode(), serverAddressPort)
35 recv()
36 duration = time() - start
37 print(f"1-4: {duration/(n*100)}")
38
39 client.close()
```

```

→ q1 git:(master) × py client.py
1-1: 0.00034749984741210937
1-2: 0.0003037810325622559
1-3: 3.136157989501953e-05
1-4: 3.278970718383789e-06
→ q1 git:(master) × py client.py
1-1: 0.0003679156303405762
1-2: 0.00033322095870971677
1-3: 4.0700435638427736e-05
1-4: 3.505229949951172e-06
→ q1 git:(master) × █

```

شکل ۱: نتایج دو بار آزمایش

توضیحات:

حالت دوم از حالت اول سریع تر است زیرا اگر رفت و برگشت i ام به اندازه x_i طول بکشد و اندازه پیام n بیت باشد و m پیام بفرستیم خواهیم داشت:

$$T_1 = n/x_1 + \dots + n/x_m$$

$$T_2 = \frac{m * n}{x_1 + \dots + x_m}$$

$$\rightarrow T_1 = n * \left(\frac{1}{x_1} + \dots + \frac{1}{x_m} \right) > n * \underbrace{\left(\frac{1}{x_1 + \dots + x_m} + \dots + \frac{1}{x_1 + \dots + x_m} \right)}_m = n * \frac{m}{x_1 + \dots + x_m} = T_2$$

البته می‌توان اشاره کرد که x_i ها در حالت اول و دوم مساوی نیستند و تاخیر صف به نفع حالت اول و در حالت دوم propagation delay کمتری داریم.

حالت سوم از حالت دوم بسیار بهتر است چرا که به جای ده بار سربار ارسال توسط سرور و کلاینت با تنها یک بار سربار همان حجم ارسال می‌شود.

و حالت چهارم از حالت سوم نیز بهتر است چرا که با مقدار سربار هندشیک و تعداد ارسال و تایید تقریباً مشابه حجم بیشتری داده را موفق به انتقال دادیم.