

# HW 1

Бабушкин Владимир Александрович, БПИ237, @babushkin05 (tg)

## цель работы

Целью работы является подсчет результатов скорости передачи данных по сети с сервера на клиента.

## оценка

Я претендую на оценку 10 (9 с учетом штрафа просрочки дедлайна на 1 день), потому что выполнил все условия. Генеративный ИИ не был использован.

## КОД

исходный код на java лежит в [./client](#) и [./server](#). Это maven проекты. Клиент создает каждый раз файл с названием **results.csv**, но я его руками переименовываю в **case1.csv** и т.д.

## КАК ПРОВОДИЛ ЭКСПЕРИМЕНТ

еще в прошлом году я собирал себе дешевенький домашний сервер ради интереса, поэтому запускал на нем

Исходный код доставил на сервер используя удаленный git репозиторий.

## Список подключенных клиентов

Тип соединения	Название устройства	Метод подключения	Сила сигнала (дБм)	Время подключения (сек)	Время аренды (сек)	IP адрес	MAC адрес	Точка подключения
Авто	yandex-mini2-6C6G	WiFi 5 ГГц	-55	11895	16905	192.168.0.12	b8:87:6e:b3:e9:1d	192.168.0.1
Авто	-	WiFi 5 ГГц	-70	4803	23997	192.168.0.39	5a:85:85:a8:27:29	192.168.0.1
Авто	-	WiFi 5 ГГц	-67	31	28769	192.168.0.42	4e:58:8e:a7:dd:ae	192.168.0.1
Авто	мои ноутбук	WiFi 5 ГГц	-59	970	27830	192.168.0.40	7a:ec:fe:31:19:b6	192.168.0.1
Авто	-	WiFi 5 ГГц	-57	2543	26257	192.168.0.43	1e:c4:02:cb:62:e7	192.168.0.1
Авто	babushkin05server	LAN 1	-	78	28722	192.168.0.30	d0:50:99:a2:59:42	192.168.0.1
Авто	-	WiFi 2.4 ГГц	-52	10278	18522	192.168.0.44	66:03:88:22:33:4f	192.168.0.1

Вот список подключенных устройств к моему роутеру

192.168.0.43 - мой ноутбук (с него запускался клиент)

192.168.0.30 - мой сервер (с него запускался сервер)

**Таблица пробрасываемых портов:**

Локальный IP адрес	Диапазон локальных портов	Протокол	Удаленный IP адрес	Диапазон удаленных портов	Wan	Статус	Комментарий	Выбрать
192.168.0.30	3659	TCP+UDP	0.0.0.0	3659	Wan1	Enabled		<input type="checkbox"/>
192.168.0.30	200-300	TCP+UDP	0.0.0.0	200-300	Wan1	Enabled		<input type="checkbox"/>
192.168.0.30	80	TCP+UDP	0.0.0.0	80	Wan1	Enabled		<input type="checkbox"/>
192.168.0.30	443	TCP+UDP	0.0.0.0	443	Wan1	Enabled		<input type="checkbox"/>
192.168.0.30	8080	TCP+UDP	0.0.0.0	8080	Wan1	Enabled		<input type="checkbox"/>

в настройках роутера добавил проброс порта 8080 на мой сервер (предыдущие пробросы я делал для сайта, ssh и другое)

таким образом на публичном ip моего роутера (я доплачиваю за статичный) по порту 8080 будет доступен сервер на джаве который я написал в предыдущем пункте.

Чтобы действовать не внутри локальной сети роутера, на своем ноутбуке я включил Нидерландский VPN, так физическое расстояние, которое проходит информация сильно увеличивается

No.	Time	Source	Destination	Protocol	Length	Info
28184	2.224.0.97	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28185	2.224.128	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28186	2.224.196	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28187	2.224.309	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28188	2.224.373	192.168.0.43	95.84.200.128	TCP	1514	65464 → 8080
28189	2.224.380	192.168.0.43	95.84.200.128	TCP	1258	65464 → 8080
28190	2.224.437	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28191	2.224.465	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28192	2.224.545	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28193	2.224.629	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28194	2.224.652	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080
28195	2.224.681	192.168.0.43	95.84.200.128	TCP	1386	65464 → 8080

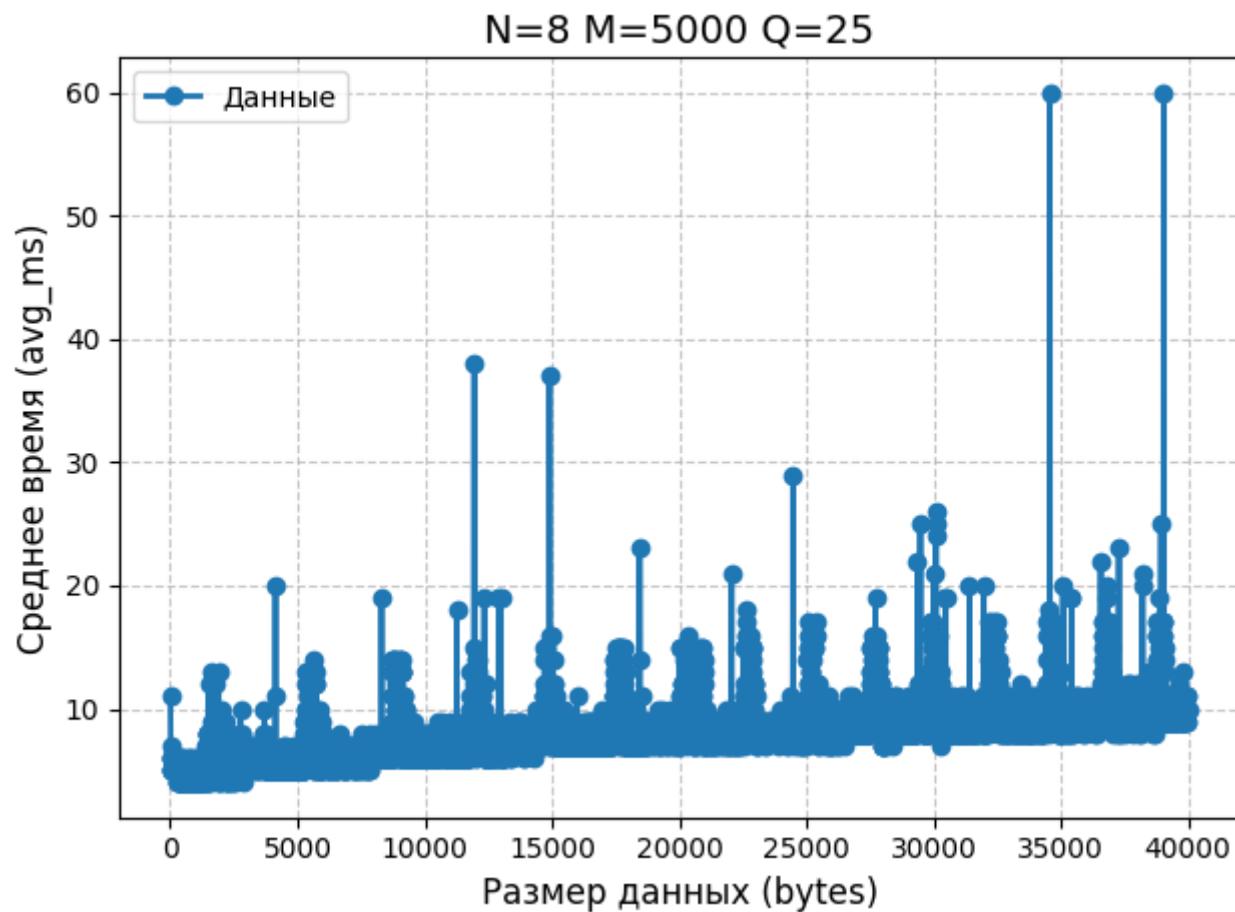
```
> Frame 28186: Packet, 1386 bytes on wire (110
> Ethernet II, Src: 1e:c4:02:cb:62:e7 (1e:c4:0
> Internet Protocol Version 4, Src: 192.168.0.
> Transmission Control Protocol, Src Port: 654
0000 f4 e5 78 9b 6c 31 1e c4 02 cb 62 e7 08 0
0010 05 5c 00 00 40 00 40 06 4c f4 c0 a8 00 0
0020 c8 80 ff b8 1f 90 19 52 43 56 05 18 cf 0
0030 08 00 c1 92 00 00 01 01 08 0a b7 95 15 0
0040 b1 fe c3 3b a1 d8 26 14 b7 19 10 bd c7 0
0050 49 91 2b 94 ca 86 2d 96 8d 2a 0a 52 e9 0
0060 0c 22 31 c1 c2 ad 3a 07 f3 23 ca 49 3b 0
0070 1b bf c9 f0 70 ac 57 e5 1f ce 64 71 32 0
0080 4f e5 71 75 23 63 33 26 c5 24 59 18 f7 0
0090 30 3d c9 79 99 41 26 a5 f4 ec 4d 80 e3 0
00a0 13 3d 69 d9 bb 78 72 ed 13 af fd 13 4d 0
00b0 19 d3 ee 32 ca b0 cb e4 ba c8 6f 99 35 0
00c0 aa 2d 56 87 7e 43 18 1a 74 e8 6a d3 a0 0
00d0 e8 59 8d fa fa be 7b bc 6c 48 28 46 1e 0
00e0 04 a7 47 89 7a b7 8e 7e 7e 31 49 13 de 0
```

скрин wireshark с клиента

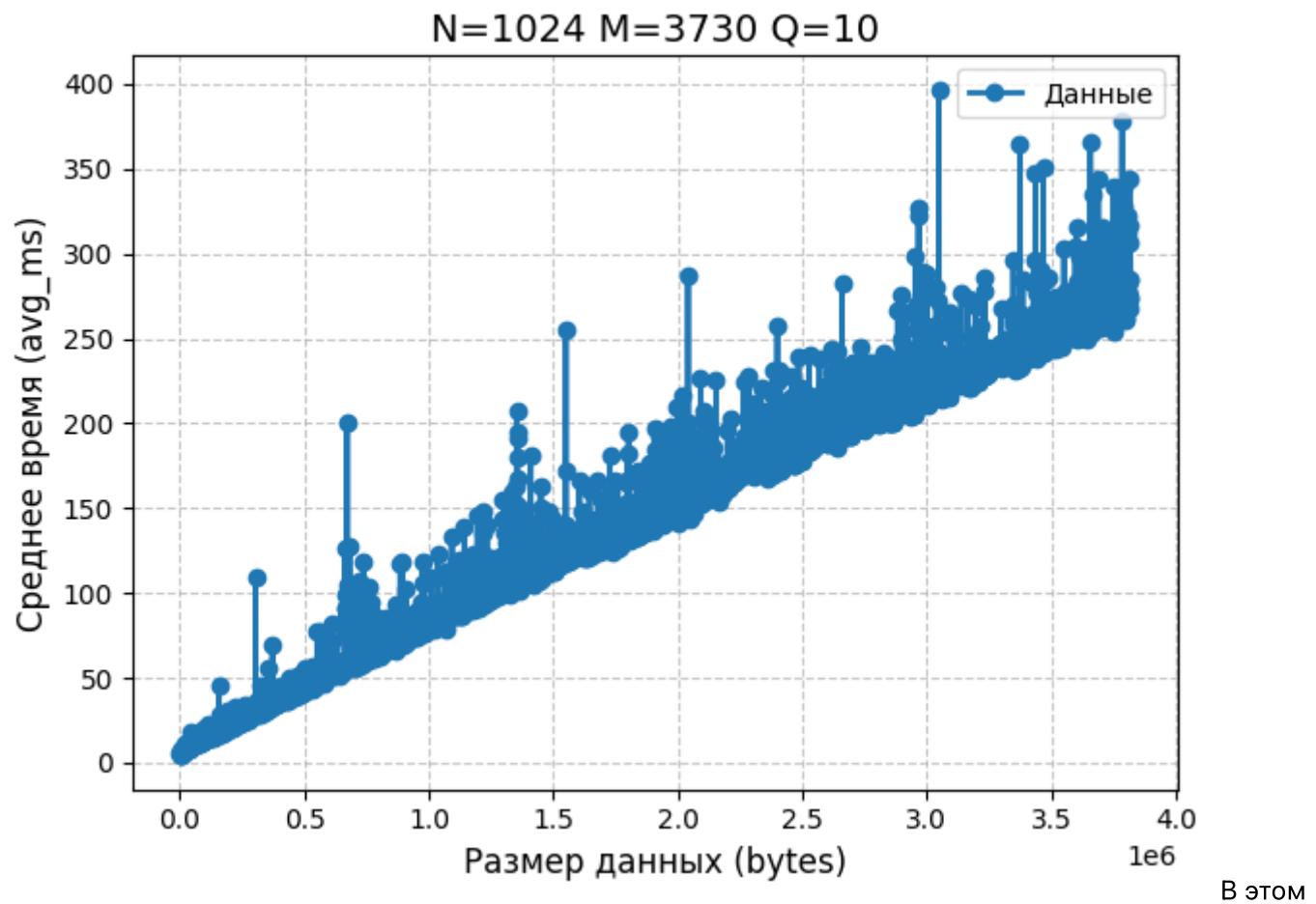
Действительно, 95.84.200.128 это ip моего сервера (можно проверить этот ip принадлежит домену babushkin05.ru)

## результаты

графики рисовал не в экселе, а в юпитер ноутбуке, мне так удобнее. Исходный код ноутбука прикреплен.

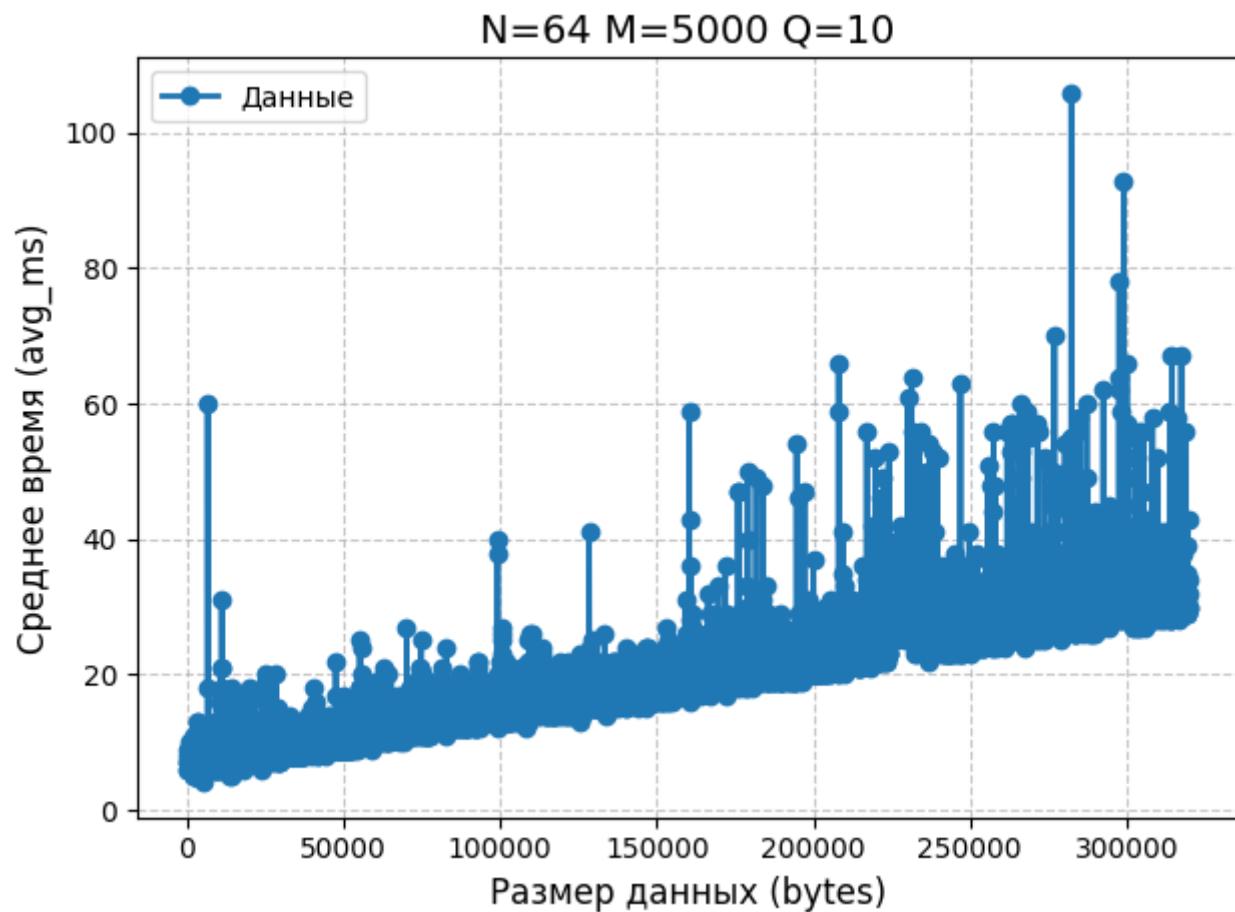


мы можем заметить много выбросов в графике, но и с ними понятен основной его принцип, при росте размера передаваемого значения, время его передачи растет медленно. Особенно заметны увеличения времени постепенное с шагом увеличивающимся в 2 раза, кажется, это связано с размерами передаваемых пакетов (они округляются вверх до степени двойки)

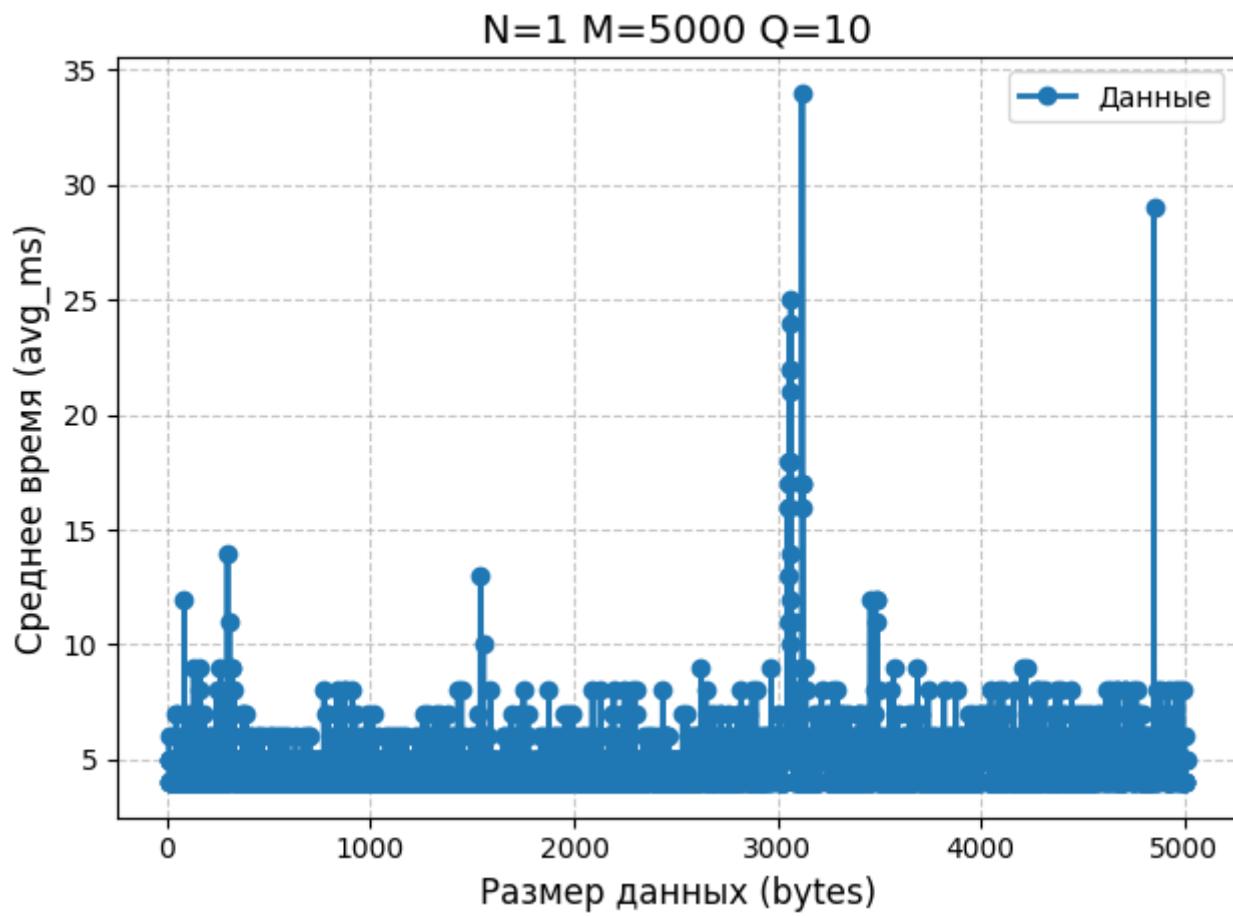


запуске считалось очень долго, поэтому я оставил работающую программу на ночь, но ночью соединение разорвалось (либо ssh соединение с сервером прекратилось и поэтому остановилась работа сервера, либо просто ноутбук ушел в спящий режим), вообщем, успело пробежать только M=3730

Но как бы то ни было, в этом случае из-за повышения размера передаваемого пакета, мы видим явную линейную зависимость времени от размера.



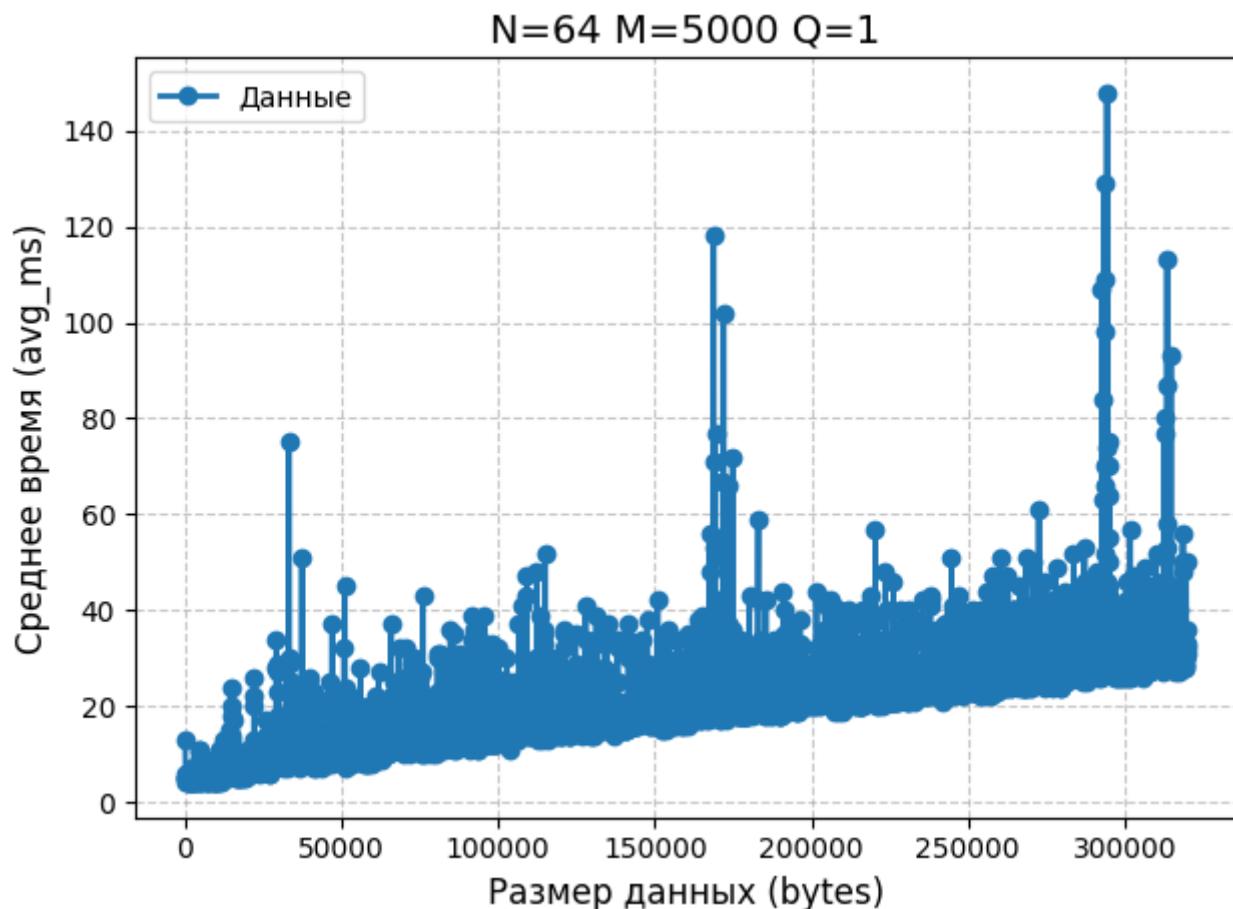
На третьем эксперименте я решил попробовать что-то среднее, и мы видим что время отправки растет, но не так сильно как при **N=1024**



теперь

решил попробовать наоборот сделать минимальный **N=1**

Как мы видим, время не растет



В последнем случае решил попробовать сделать **Q=1**, получился результат идентичный третьему случаю, получается округление особо не имело смысла.