Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Лабораторная работа №4

Дисциплина: «Моделирование работы канала»

Тема: «Стохастические сетевые модели вычислительных систем»

Вариант 2

Выполнил

Студент группы УПАСбд-31

Джураев И.Д.

Проверил

преподаватель кафедры

«Вычислительная техника»

Валюх В. В.

Ульяновск, 2024

**Цель работы.** Изучение характеристик работы канала, моделирование его работы. Изучение влияния скорости передачи информации и загрузки канала на его пропускную способность.

ЗАДАНИЕ 1

Для потока заявок, полученного в лабораторной работе 3, выполнить моделирование работы канала для нескольких вариантов его скорости передачи информации. Вычислить характеристики входного потока заявок (), очереди заявок работающего канала (). Построить графики зависимости характеристик от скорости работы канала. Считать дисциплину обслуживания потока заявок каналом бесприоритетной.

ЗАДАНИЕ 2

Выполнить моделирование работы канала и вычисление его характеристик для дисциплины обслуживания, определяемой матрицей приоритетов варианта задания.

**ОПИСАНИЕ РЕАЛИЗАЦИИ**

Для моделирования канала, надо учитывать время, потраченное на обработку каждого сообщения, и, если следующее сообщение приходит раньше, чем обработалось предыдущее, нужно организовать очередь.

Для очереди без приоритетов, мы будем вычислять время окончания обработки текущего элемента. Затем, на следующей итерации, будем вычитать из этого времени, время поступления элемента, и, если разность будет больше 0, то время поступления увеличиваем на данную разность

Для создания приоритетной очереди, надо добавить запись всей последовательности в массив, а также создать условие, что если приоритет текущего элемента, выше, чем приоритет прошлого, и его время возникновения раньше, чем время появления прошлого элемента, с учетом очереди, то тогда они меняются в массиве местами и меняется их время возникновения с учетом очереди

Для подсчета характеристик, в цикле создания очереди, будем суммировать по типам: время обслуживания, количество заявок, время между поступлением заявок, время в очереди, первый и второй начальные моменты.

Затем, по формулам из теории, находим все необходимые характеристики (таблица 1, 2).

– средняя скорость обслуживания заявки -го типа;

– второй начальный момент времени обслуживания;

– дисперсия времени обслуживания;

– среднеквадратичное отклонение времени обслуживания;

– интенсивность обслуживания;

– средний промежуток времени между поступлением заявок i-го типа;

– интенсивность поступления заявок -го типа;

– коэффициент загрузки оборудования заявками-го типа;

– коэффициент простоя;

– коэффициент загрузки;

– интенсивность поступления заявок;

– вероятность поступления заявки -го типа;

– среднее время пребывания заявки-го типа в очереди;

– среднее время пребывания заявки в системе

– средняя длина очереди заявок -го типа;

– среднее время пребывания заявки в очереди;

= – среднее время пребывания заявки в системе;

– среднее число заявок в системы.

Таблица 1 **–** очередь без приоритета

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Параметр** | **Тип 1, скорость 40** | **Тип 2, скорость 40** | **Тип 3, скорость 40** | **Тип 4, скорость 40** | **Тип 1, скорость 50** | **Тип 2, скорость 50** | **Тип 3, скорость 50** | **Тип 4, скорость 50** | **Тип 1, скорость 60** | **Тип 2, скорость 60** | **Тип 3, скорость 60** | **Тип 4, скорость 60** |
|  | 2,2308 | 2,1017 | 2,1429 | 1,9048 | 1,6154 | 1,4576 | 1,5714 | 1,5238 | 1,3846 | 1,2203 | 1,1429 | 1 |
|  | 33877,425 | 1,0755E+19 | 207,4687 | 752756,6979 | 510,0201 | 4520015629 | 23,6626 | 6942,9551 | 68,751 | 126575,1943 | 2,5465 | 1 |
|  | 1,4083 | 1,5829 | 1,2653 | 2,2766 | 1,0059 | 1,1296 | 0,5306 | 1,3923 | 1,0059 | 0,9176 | 0,6939 | 0,6667 |
|  | 1,1867 | 1,2581 | 1,1249 | 1,5088 | 1,0029 | 1,0628 | 0,7284 | 1,18 | 1,0029 | 0,9579 | 0,833 | 0,8165 |
|  | 0,4483 | 0,4758 | 0,4667 | 0,525 | 0,619 | 0,686 | 0,6364 | 0,6562 | 0,7222 | 0,8194 | 0,875 | 1 |
|  | 0,0793 | 0,3642 | 0,0504 | 0,1288 | 0,0793 | 0,3642 | 0,0504 | 0,1288 | 0,0793 | 0,3642 | 0,0504 | 0,1288 |
|  | 0,1769 | 0,7654 | 0,108 | 0,2453 | 0,1281 | 0,5309 | 0,0792 | 0,1963 | 0,1098 | 0,4445 | 0,0576 | 0,1288 |
|  | 0,8231 | 0,2346 | 0,892 | 0,7547 | 0,8719 | 0,4691 | 0,9208 | 0,8037 | 0,8902 | 0,5555 | 0,9424 | 0,8712 |
|  | 12,6154 | 2,7458 | 19,8571 | 7,7619 | 12,6154 | 2,7458 | 19,8571 | 7,7619 | 12,6154 | 2,7458 | 19,8571 | 7,7619 |
|  | 0,127348643 | 0,58487233 | 0,080937851 | 0,206841176 | 0,127348643 | 0,58487233 | 0,080937851 | 0,206841176 | 0,127348643 | 0,58487233 | 0,080937851 | 0,206841176 |
|  | 11,3846 | 16,5085 | 21,7143 | 16,5238 | 1,1538 | 2,6271 | 3,4286 | 2,2857 | 0,3846 | 0,5763 | 0,2857 | 0,5714 |
|  | 13,6154 | 18,6102 | 23,8571 | 18,4286 | 2,7692 | 4,0847 | 5 | 3,8095 | 1,7692 | 1,7966 | 1,4286 | 1,5714 |
|  | 0,9028 | 6,0124 | 1,0944 | 2,1283 | 0,0915 | 0,9568 | 0,1728 | 0,2944 | 0,0305 | 0,2099 | 0,0144 | 0,0736 |
|  | 1,2956 | | | | 0,9345 | | | | 0,7407 | | | |
|  | 0,6227 | | | | 0,6227 | | | | 0,6227 | | | |
| W | 16,28048922 | | | | 2,433733355 | | | | 0,527353204 | | | |
| U | 18,36122945 | | | | 3,934332584 | | | | 1,716744885 | | | |
| L | 11,43353758 | | | | 2,4499089 | | | | 1,06901704 | | | |

Таблица 2 **–** очередь с приоритетом

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Параметр** | **Тип 1, скорость 40** | **Тип 2, скорость 40** | **Тип 3, скорость 40** | **Тип 4, скорость 40** | **Тип 1, скорость 50** | **Тип 2, скорость 50** | **Тип 3, скорость 50** | **Тип 4, скорость 50** | **Тип 1, скорость 60** | **Тип 2, скорость 60** | **Тип 3, скорость 60** | **Тип 4, скорость 60** |
|  | 2,2308 | 2,1017 | 2,1429 | 1,9048 | 1,6154 | 1,4576 | 1,5714 | 1,5238 | 1,3846 | 1,2203 | 1,1429 | 1 |
|  | 33877,425 | 1,0755E+19 | 207,4687 | 752756,6979 | 510,0201 | 4520015629 | 23,6626 | 6942,9551 | 68,751 | 126575,1943 | 2,5465 | 1 |
|  | 1,4083 | 1,5829 | 1,2653 | 2,2766 | 1,0059 | 1,1296 | 0,5306 | 1,3923 | 1,0059 | 0,9176 | 0,6939 | 0,6667 |
|  | 1,1867 | 1,2581 | 1,1249 | 1,5088 | 1,0029 | 1,0628 | 0,7284 | 1,18 | 1,0029 | 0,9579 | 0,833 | 0,8165 |
|  | 0,4483 | 0,4758 | 0,4667 | 0,525 | 0,619 | 0,686 | 0,6364 | 0,6562 | 0,7222 | 0,8194 | 0,875 | 1 |
|  | 0,0793 | 0,3642 | 0,0504 | 0,1288 | 0,0793 | 0,3642 | 0,0504 | 0,1288 | 0,0793 | 0,3642 | 0,0504 | 0,1288 |
|  | 0,1769 | 0,7654 | 0,108 | 0,2453 | 0,1281 | 0,5309 | 0,0792 | 0,1963 | 0,1098 | 0,4445 | 0,0576 | 0,1288 |
|  | 0,8231 | 0,2346 | 0,892 | 0,7547 | 0,8719 | 0,4691 | 0,9208 | 0,8037 | 0,8902 | 0,5555 | 0,9424 | 0,8712 |
|  | 12,6154 | 2,7458 | 19,8571 | 7,7619 | 12,6154 | 2,7458 | 19,8571 | 7,7619 | 12,6154 | 2,7458 | 19,8571 | 7,7619 |
|  | 0,127348643 | 0,58487233 | 0,080937851 | 0,206841176 | 0,127348643 | 0,58487233 | 0,080937851 | 0,206841176 | 0,127348643 | 0,58487233 | 0,080937851 | 0,206841176 |
|  | 1,8462 | 5,3729 | 25,8571 | 60,9048 | 0,5385 | 1,4746 | 1,4286 | 6,0476 | 0,3846 | 0,5763 | 0,2857 | 0,5714 |
|  | 4,0769 | 7,4746 | 28 | 62,8095 | 2,1538 | 2,9322 | 3 | 7,5714 | 1,7692 | 1,7966 | 1,4286 | 1,5714 |
|  | 0,1464 | 1,9568 | 1,3032 | 7,8445 | 0,0427 | 0,537 | 0,072 | 0,7789 | 0,0305 | 0,2099 | 0,0144 | 0,0736 |
|  | 1,2956 | | | | 0,9345 | | | | 0,7407 | | | |
|  | 0,6227 | | | | 0,6227 | | | | 0,6227 | | | |
| W | 18,06801015 | | | | 2,29755049 | | | | 0,527353204 | | | |
| U | 20,14872505 | | | | 3,798136984 | | | | 1,716744885 | | | |
| L | 12,54661109 | | | | 2,3650999 | | | | 1,06901704 | | | |

По найденным характеристикам, построим графики зависимости характеристики от скорости обслуживания

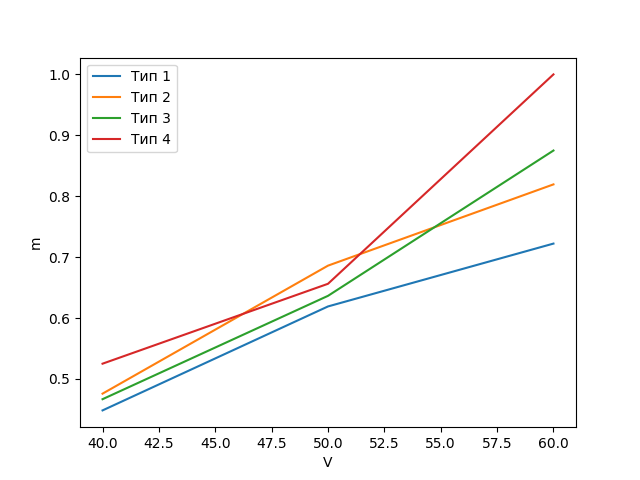


Рис. 1 – Зависимость m от V

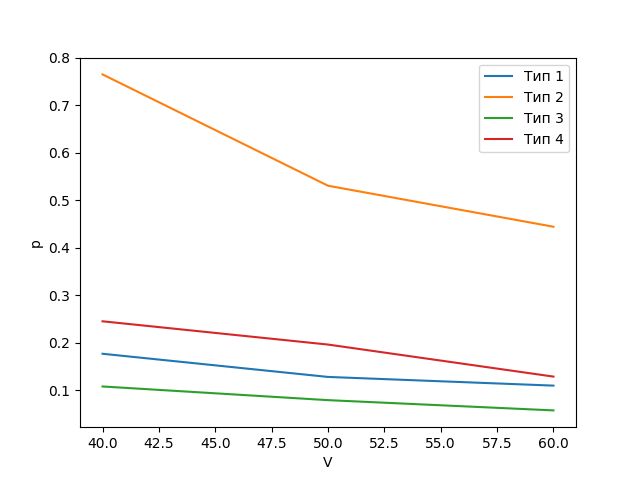


Рис. 2 – Зависимость p от V

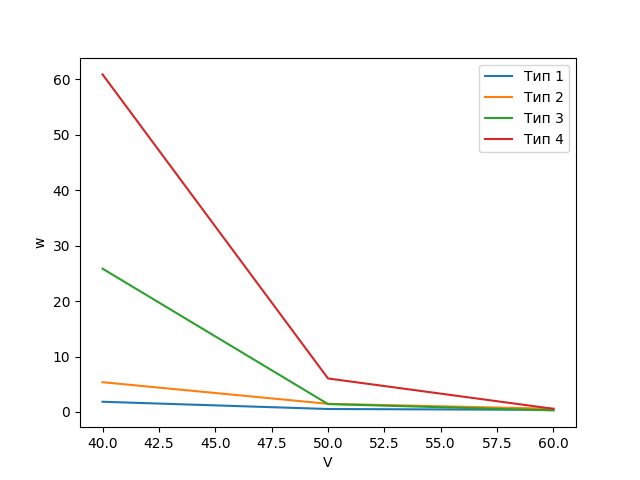


Рис. 3 – Зависимость w от V

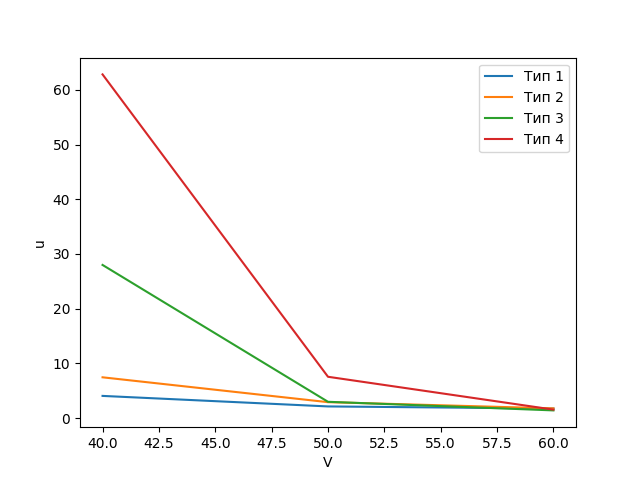


Рис4 – Зависимость u от V

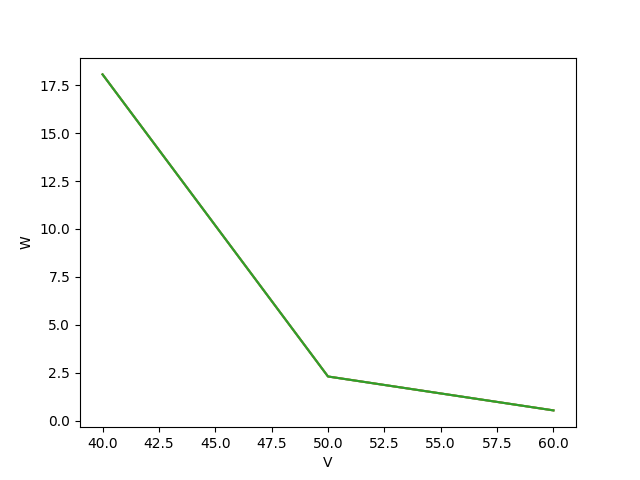


Рис. 5 – Зависимость W от V

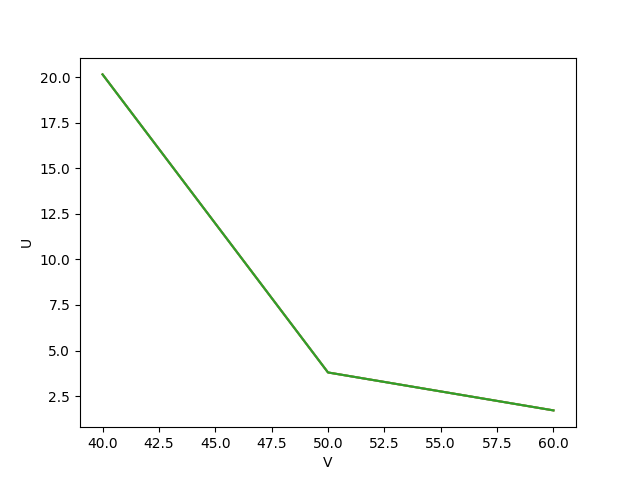


Рис. 6 – Зависимость U от V

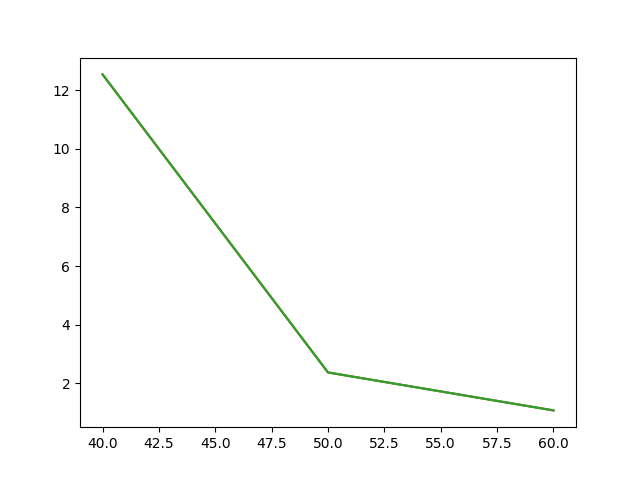


Рис. 7 – Зависимость L от V

Наиболее оптимальный по уровню загрузки и скорости передачи информации, канал, имеющий скорость v = 60. Получим для него распечатку процесса моделирования (таблица 3 для очереди без приоритета и таблица 4 для очереди с приоритетом).

Таблица 3 – Процесс моделирования для очереди без приоритета.

| **Номер заявки** | **Тип заявки** | **Момент появления** | **Начало обслуживания** | **Конец обслуживания** | **Ожидание** | **Время пребывания в канале** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 0 | 0 | 2 | 0 | 2 |
| 2 | 4 | 2 | 2 | 4 | 0 | 2 |
| 3 | 2 | 4 | 4 | 6 | 0 | 2 |
| 4 | 1 | 5 | 6 | 7 | 1 | 2 |
| 5 | 2 | 7 | 7 | 9 | 0 | 2 |
| 6 | 1 | 10 | 10 | 10 | 0 | 0 |
| 7 | 1 | 11 | 11 | 14 | 0 | 3 |
| 8 | 1 | 12 | 14 | 15 | 2 | 3 |
| 9 | 2 | 13 | 15 | 16 | 2 | 3 |
| 10 | 4 | 15 | 16 | 17 | 1 | 2 |
| 11 | 2 | 16 | 17 | 19 | 1 | 3 |
| 12 | 2 | 17 | 19 | 20 | 2 | 3 |
| 13 | 4 | 20 | 20 | 20 | 0 | 0 |
| 14 | 1 | 21 | 21 | 22 | 0 | 1 |
| 15 | 2 | 22 | 22 | 22 | 0 | 0 |
| 16 | 2 | 23 | 23 | 23 | 0 | 0 |
| 17 | 1 | 25 | 25 | 28 | 0 | 3 |
| 18 | 2 | 28 | 28 | 30 | 0 | 2 |
| 19 | 3 | 29 | 30 | 32 | 1 | 3 |
| 20 | 2 | 30 | 32 | 33 | 2 | 3 |
| 21 | 2 | 33 | 33 | 34 | 0 | 1 |
| 22 | 4 | 35 | 35 | 37 | 0 | 2 |
| 23 | 1 | 38 | 38 | 39 | 0 | 1 |
| 24 | 2 | 39 | 39 | 39 | 0 | 0 |
| 25 | 2 | 41 | 41 | 41 | 0 | 0 |
| 26 | 2 | 43 | 43 | 45 | 0 | 2 |
| 27 | 2 | 46 | 46 | 46 | 0 | 0 |
| 28 | 1 | 47 | 47 | 48 | 0 | 1 |
| 29 | 2 | 48 | 48 | 49 | 0 | 1 |
| 30 | 1 | 50 | 50 | 51 | 0 | 1 |
| 31 | 4 | 52 | 52 | 54 | 0 | 2 |
| 32 | 2 | 54 | 54 | 57 | 0 | 3 |
| 33 | 2 | 55 | 57 | 57 | 2 | 2 |
| 34 | 2 | 56 | 57 | 58 | 1 | 2 |
| 35 | 2 | 57 | 58 | 59 | 1 | 2 |
| 36 | 2 | 60 | 60 | 61 | 0 | 1 |
| 37 | 2 | 63 | 63 | 64 | 0 | 1 |
| 38 | 4 | 65 | 65 | 66 | 0 | 1 |
| 39 | 2 | 66 | 66 | 67 | 0 | 1 |
| 40 | 2 | 68 | 68 | 71 | 0 | 3 |
| 41 | 2 | 70 | 71 | 71 | 1 | 1 |
| 42 | 2 | 71 | 71 | 72 | 0 | 1 |
| 43 | 4 | 73 | 73 | 73 | 0 | 0 |
| 44 | 2 | 74 | 74 | 76 | 0 | 2 |
| 45 | 4 | 75 | 76 | 78 | 1 | 3 |
| 46 | 2 | 76 | 78 | 78 | 2 | 2 |
| 47 | 2 | 77 | 78 | 78 | 1 | 1 |
| 48 | 2 | 78 | 78 | 78 | 0 | 0 |
| 49 | 4 | 80 | 80 | 80 | 0 | 0 |
| 50 | 2 | 82 | 82 | 82 | 0 | 0 |
| 51 | 3 | 84 | 84 | 85 | 0 | 1 |
| 52 | 2 | 85 | 85 | 87 | 0 | 2 |
| 53 | 2 | 87 | 87 | 87 | 0 | 0 |
| 54 | 3 | 88 | 88 | 88 | 0 | 0 |
| 55 | 1 | 90 | 90 | 93 | 0 | 3 |
| 56 | 2 | 92 | 93 | 96 | 1 | 4 |
| 57 | 2 | 93 | 96 | 96 | 3 | 3 |
| 58 | 4 | 96 | 96 | 96 | 0 | 0 |
| 59 | 2 | 97 | 97 | 98 | 0 | 1 |
| 60 | 4 | 98 | 98 | 99 | 0 | 1 |
| 61 | 4 | 101 | 101 | 101 | 0 | 0 |
| 62 | 2 | 104 | 104 | 104 | 0 | 0 |
| 63 | 4 | 107 | 107 | 108 | 0 | 1 |
| 64 | 2 | 108 | 108 | 108 | 0 | 0 |
| 65 | 2 | 110 | 110 | 112 | 0 | 2 |
| 66 | 2 | 113 | 113 | 115 | 0 | 2 |
| 67 | 4 | 114 | 115 | 115 | 1 | 1 |
| 68 | 1 | 116 | 116 | 117 | 0 | 1 |
| 69 | 4 | 117 | 117 | 119 | 0 | 2 |
| 70 | 2 | 118 | 119 | 121 | 1 | 3 |
| 71 | 2 | 120 | 121 | 122 | 1 | 2 |
| 72 | 4 | 121 | 122 | 123 | 1 | 2 |
| 73 | 2 | 123 | 123 | 125 | 0 | 2 |
| 74 | 2 | 125 | 125 | 125 | 0 | 0 |
| 75 | 2 | 126 | 126 | 128 | 0 | 2 |
| 76 | 2 | 128 | 128 | 129 | 0 | 1 |
| 77 | 3 | 129 | 129 | 131 | 0 | 2 |
| 78 | 2 | 130 | 131 | 133 | 1 | 3 |
| 79 | 1 | 131 | 133 | 133 | 2 | 2 |
| 80 | 2 | 132 | 133 | 135 | 1 | 3 |
| 81 | 4 | 133 | 135 | 137 | 2 | 4 |
| 82 | 4 | 135 | 137 | 139 | 2 | 4 |
| 83 | 2 | 137 | 139 | 141 | 2 | 4 |
| 84 | 2 | 139 | 141 | 144 | 2 | 5 |
| 85 | 2 | 141 | 144 | 145 | 3 | 4 |
| 86 | 3 | 144 | 145 | 145 | 1 | 1 |
| 87 | 3 | 146 | 146 | 148 | 0 | 2 |
| 88 | 2 | 147 | 148 | 149 | 1 | 2 |
| 89 | 2 | 148 | 149 | 150 | 1 | 2 |
| 90 | 2 | 152 | 152 | 154 | 0 | 2 |
| 91 | 2 | 153 | 154 | 155 | 1 | 2 |
| 92 | 2 | 154 | 155 | 156 | 1 | 2 |
| 93 | 4 | 155 | 156 | 156 | 1 | 1 |
| 94 | 2 | 156 | 156 | 159 | 0 | 3 |
| 95 | 2 | 159 | 159 | 160 | 0 | 1 |
| 96 | 2 | 162 | 162 | 165 | 0 | 3 |
| 97 | 4 | 163 | 165 | 166 | 2 | 3 |
| 98 | 4 | 165 | 166 | 167 | 1 | 2 |
| 99 | 3 | 168 | 168 | 169 | 0 | 1 |
| 100 | 1 | 169 | 169 | 171 | 0 | 2 |

Таблица 4 – Процесс моделирования для очереди с приоритетом.

| **Номер заявки** | **Тип заявки** | **Момент появления** | **Начало обслуживания** | **Конец обслуживания** | **Ожидание** | **Время пребывания в канале** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 0 | 0 | 2 | 0 | 2 |
| 2 | 4 | 2 | 2 | 4 | 0 | 2 |
| 3 | 2 | 4 | 4 | 6 | 0 | 2 |
| 4 | 1 | 5 | 6 | 7 | 1 | 2 |
| 5 | 2 | 7 | 7 | 9 | 0 | 2 |
| 6 | 1 | 10 | 10 | 10 | 0 | 0 |
| 7 | 1 | 11 | 11 | 14 | 0 | 3 |
| 8 | 1 | 12 | 14 | 15 | 2 | 3 |
| 9 | 2 | 13 | 15 | 16 | 2 | 3 |
| 10 | 4 | 15 | 16 | 17 | 1 | 2 |
| 11 | 2 | 16 | 17 | 19 | 1 | 3 |
| 12 | 2 | 17 | 19 | 20 | 2 | 3 |
| 13 | 4 | 20 | 20 | 20 | 0 | 0 |
| 14 | 1 | 21 | 21 | 22 | 0 | 1 |
| 15 | 2 | 22 | 22 | 22 | 0 | 0 |
| 16 | 2 | 23 | 23 | 23 | 0 | 0 |
| 17 | 1 | 25 | 25 | 28 | 0 | 3 |
| 18 | 2 | 28 | 28 | 30 | 0 | 2 |
| 19 | 3 | 29 | 30 | 32 | 1 | 3 |
| 20 | 2 | 30 | 32 | 33 | 2 | 3 |
| 21 | 2 | 33 | 33 | 34 | 0 | 1 |
| 22 | 4 | 35 | 35 | 37 | 0 | 2 |
| 23 | 1 | 38 | 38 | 39 | 0 | 1 |
| 24 | 2 | 39 | 39 | 39 | 0 | 0 |
| 25 | 2 | 41 | 41 | 41 | 0 | 0 |
| 26 | 2 | 43 | 43 | 45 | 0 | 2 |
| 27 | 2 | 46 | 46 | 46 | 0 | 0 |
| 28 | 1 | 47 | 47 | 48 | 0 | 1 |
| 29 | 2 | 48 | 48 | 49 | 0 | 1 |
| 30 | 1 | 50 | 50 | 51 | 0 | 1 |
| 31 | 4 | 52 | 52 | 54 | 0 | 2 |
| 32 | 2 | 54 | 54 | 57 | 0 | 3 |
| 33 | 2 | 55 | 57 | 57 | 2 | 2 |
| 34 | 2 | 56 | 57 | 58 | 1 | 2 |
| 35 | 2 | 57 | 58 | 59 | 1 | 2 |
| 36 | 2 | 60 | 60 | 61 | 0 | 1 |
| 37 | 2 | 63 | 63 | 64 | 0 | 1 |
| 38 | 4 | 65 | 65 | 66 | 0 | 1 |
| 39 | 2 | 66 | 66 | 67 | 0 | 1 |
| 40 | 2 | 68 | 68 | 71 | 0 | 3 |
| 41 | 2 | 70 | 71 | 71 | 1 | 1 |
| 42 | 2 | 71 | 71 | 72 | 0 | 1 |
| 43 | 4 | 73 | 73 | 73 | 0 | 0 |
| 44 | 2 | 74 | 74 | 76 | 0 | 2 |
| 45 | 4 | 75 | 76 | 78 | 1 | 3 |
| 46 | 2 | 76 | 78 | 78 | 2 | 2 |
| 47 | 2 | 77 | 78 | 78 | 1 | 1 |
| 48 | 2 | 78 | 78 | 78 | 0 | 0 |
| 49 | 4 | 80 | 80 | 80 | 0 | 0 |
| 50 | 2 | 82 | 82 | 82 | 0 | 0 |
| 51 | 3 | 84 | 84 | 85 | 0 | 1 |
| 52 | 2 | 85 | 85 | 87 | 0 | 2 |
| 53 | 2 | 87 | 87 | 87 | 0 | 0 |
| 54 | 3 | 88 | 88 | 88 | 0 | 0 |
| 55 | 1 | 90 | 90 | 93 | 0 | 3 |
| 56 | 2 | 92 | 93 | 96 | 1 | 4 |
| 57 | 2 | 93 | 96 | 96 | 3 | 3 |
| 58 | 4 | 96 | 96 | 96 | 0 | 0 |
| 59 | 2 | 97 | 97 | 98 | 0 | 1 |
| 60 | 4 | 98 | 98 | 99 | 0 | 1 |
| 61 | 4 | 101 | 101 | 101 | 0 | 0 |
| 62 | 2 | 104 | 104 | 104 | 0 | 0 |
| 63 | 4 | 107 | 107 | 108 | 0 | 1 |
| 64 | 2 | 108 | 108 | 108 | 0 | 0 |
| 65 | 2 | 110 | 110 | 112 | 0 | 2 |
| 66 | 2 | 113 | 113 | 115 | 0 | 2 |
| 67 | 4 | 114 | 115 | 115 | 1 | 1 |
| 68 | 1 | 116 | 116 | 117 | 0 | 1 |
| 69 | 4 | 117 | 117 | 119 | 0 | 2 |
| 70 | 2 | 118 | 119 | 121 | 1 | 3 |
| 71 | 2 | 120 | 121 | 122 | 1 | 2 |
| 72 | 4 | 121 | 122 | 123 | 1 | 2 |
| 73 | 2 | 123 | 123 | 125 | 0 | 2 |
| 74 | 2 | 125 | 125 | 125 | 0 | 0 |
| 75 | 2 | 126 | 126 | 128 | 0 | 2 |
| 76 | 2 | 128 | 128 | 129 | 0 | 1 |
| 77 | 3 | 129 | 129 | 131 | 0 | 2 |
| 78 | 2 | 130 | 131 | 133 | 1 | 3 |
| 79 | 1 | 131 | 133 | 133 | 2 | 2 |
| 80 | 2 | 132 | 133 | 135 | 1 | 3 |
| 81 | 4 | 133 | 135 | 137 | 2 | 4 |
| 82 | 4 | 135 | 137 | 139 | 2 | 4 |
| 83 | 2 | 137 | 139 | 141 | 2 | 4 |
| 84 | 2 | 139 | 141 | 144 | 2 | 5 |
| 85 | 2 | 141 | 144 | 145 | 3 | 4 |
| 86 | 3 | 144 | 145 | 145 | 1 | 1 |
| 87 | 3 | 146 | 146 | 148 | 0 | 2 |
| 88 | 2 | 147 | 148 | 149 | 1 | 2 |
| 89 | 2 | 148 | 149 | 150 | 1 | 2 |
| 90 | 2 | 152 | 152 | 154 | 0 | 2 |
| 91 | 2 | 153 | 154 | 155 | 1 | 2 |
| 92 | 2 | 154 | 155 | 156 | 1 | 2 |
| 93 | 4 | 155 | 156 | 156 | 1 | 1 |
| 94 | 2 | 156 | 156 | 159 | 0 | 3 |
| 95 | 2 | 159 | 159 | 160 | 0 | 1 |
| 96 | 2 | 162 | 162 | 165 | 0 | 3 |
| 97 | 4 | 163 | 165 | 166 | 2 | 3 |
| 98 | 4 | 165 | 166 | 167 | 1 | 2 |
| 99 | 3 | 168 | 168 | 169 | 0 | 1 |
| 100 | 1 | 169 | 169 | 171 | 0 | 2 |

**Выводы о проделанной работе**

Выполняя данную лабораторную работу, мы изучили характеристики канала, научились его моделировать. Изучили влияния скорости передачи информации и загрузки канала на его пропускную способность.

**main.py**

|  |
| --- |
| from config import \*  from messages import Message  import math  import statistics  import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  class Stream:  velocity: int  flow: list  stream\_data: list  stream\_data\_by\_type: dict  queue: list  finish\_time: int  start\_time: int  def \_\_init\_\_(self, flow: list[Message], velocity: int, is\_priority: bool = False):  self.flow = flow  self.velocity = velocity  self.queue = []  self.stream\_data = []  self.stream\_data\_by\_type = {i: [] for i in TYPES}  self.finish\_time = 0  self.start\_time = 0  self.imitate\_stream(is\_priority=is\_priority)  def order\_queue(self):  self.queue = sorted(self.queue, key=lambda x: x[0].ms\_type)  def imitate\_stream(self, is\_priority: bool):  for stream\_time in range(1000):  if len(self.queue) != 0 and stream\_time == self.queue[0][1]:  if len(self.queue) != 0:  while stream\_time == self.queue[0][1]:  # if is\_priority:  # self.order\_queue()  self.calculate\_stream\_data(self.queue[0][0])  self.queue.pop(0)  if len(self.queue) == 0:  break  for message in self.queue:  message[1] = self.finish\_time  for ind, message in enumerate(self.flow):  if message.timestamp > stream\_time:  break  if message.timestamp == stream\_time:  if message.timestamp < self.finish\_time:  self.queue.append([message, self.finish\_time])  self.flow.pop(ind)  if message.timestamp == self.finish\_time:  self.calculate\_stream\_data(message)  self.flow.pop(ind)  if message.timestamp > self.finish\_time:  self.calculate\_stream\_data(message, stream\_awaiting=message.timestamp - self.finish\_time)  self.flow.pop(ind)  def calculate\_stream\_data(self, message: Message, stream\_awaiting: int = 0):  appearance\_time = message.timestamp  start\_time = message.timestamp  awaiting\_time = self.finish\_time - start\_time  if awaiting\_time > 0:  start\_time += awaiting\_time  else:  awaiting\_time = 0  self.finish\_time = int(message.message\_len / self.velocity + start\_time)  self.stream\_data.append({"type": message.ms\_type,  "len": message.message\_len,  "appearance\_time": appearance\_time,  "start\_time": start\_time,  "finish\_time": self.finish\_time,  "total\_time": self.finish\_time - start\_time,  "awaiting\_time": awaiting\_time,  "stream\_awaiting": stream\_awaiting})  self.stream\_data\_by\_type[message.ms\_type].append({"len": message.message\_len,  "appearance\_time": appearance\_time,  "start\_time": start\_time,  "finish\_time": self.finish\_time,  "total\_time": self.finish\_time - start\_time,  "awaiting\_time": awaiting\_time,  "stream\_awaiting": stream\_awaiting})  def get\_stream\_data(self):  fields = ["Номер заявки", "Тип заявки", "Момент появления",  "Начало обслуживания", "Конец обслуживания", "Ожидание", "Время прибывания в канале"]  res = {i: {} for i in fields}  for i, val in enumerate(self.stream\_data):  values = [i + 1, val["type"], val["appearance\_time"], val["start\_time"],  val["finish\_time"], val["awaiting\_time"], val["finish\_time"] - val["appearance\_time"]]  for n, j in enumerate(res):  res[j][i] = values[n]  return res  params = ["v", "v^n", "D", "o", "m", "la", "p", "n", "t", "p2", "w", "u", "l"]  def calculate\_stats(data: dict) -> tuple[dict, dict]:  total\_type\_stats = {"R": 0, "/|": 0, "W": 0, "U": 0, "L": 0}  type\_stats = {i: {j: None for j in params} for i in TYPES}  for t in TYPES:  messages = data[t]  times = [i["total\_time"] for i in messages]  v = sum(times) / len(messages)  type\_stats[t]["v"] = round(v, 4)  type\_stats[t]["v^n"] = round(v \*\* len(messages), 4)  type\_stats[t]["D"] = round(statistics.pvariance(times), 4)  type\_stats[t]["o"] = round(math.sqrt(type\_stats[t]["D"]), 4)  type\_stats[t]["m"] = round(1 / v, 4)  type\_stats[t]["t"] = round(sum([i["appearance\_time"] for i in messages]) / len(messages), 4)  type\_stats[t]["t"] = round(sum(np.diff([i["appearance\_time"] for i in messages])) / len(messages), 4)    type\_stats[t]["la"] = round(1 / type\_stats[t]["t"], 4)  type\_stats[t]["p"] = round(type\_stats[t]["la"] / type\_stats[t]["m"], 4)  type\_stats[t]["n"] = round(1 - type\_stats[t]["p"], 4)  total\_type\_stats["R"] += round(type\_stats[t]["p"], 4)  total\_type\_stats["/|"] += round(type\_stats[t]["la"], 4)  type\_stats[t]["w"] = round(sum([i["awaiting\_time"] for i in messages]) / len(messages), 4)  type\_stats[t]["u"] = round(sum([i["awaiting\_time"] + i["total\_time"] for i in messages]) / len(messages), 4)  type\_stats[t]["l"] = round(type\_stats[t]["w"] \* type\_stats[t]["la"], 4)  for t in TYPES:  type\_stats[t]["p2"] = type\_stats[t]["la"] / total\_type\_stats['/|']  total\_type\_stats["W"] += type\_stats[t]["p2"] \* type\_stats[t]["w"]  total\_type\_stats["U"] += type\_stats[t]["p2"] \* type\_stats[t]["u"]  total\_type\_stats["L"] += type\_stats[t]["la"] \* type\_stats[t]["u"]  return type\_stats, total\_type\_stats  def to\_excel(type\_stats: list, total\_stats: list, stream\_data: dict):  total\_params = ["R", "/|", "W", "U", "L"]  data = {"Параметр": {i: i for i in params + total\_params}}  for i in range(len(type\_stats)):  sub\_data: dict = type\_stats[i]  for j in range(len(sub\_data)):  sub\_data[f"Тип {j + 1}, скорость {VELOCITIES[i]}"] = sub\_data[j + 1]  sub\_data.pop(j + 1)  sub\_data[f"Тип 1, скорость {VELOCITIES[i]}"].update(total\_stats[i])  data.update(sub\_data)  data.update({"": ""})  data.update(stream\_data)  df = pd.DataFrame(data)  pd.options.display.float\_format = '{:,.4}'.format  df.to\_excel("output\_data1.xlsx", index=False, engine="openpyxl")  if \_\_name\_\_ == '\_\_main\_\_':  all\_type\_stats = []  all\_total\_stats = []  final\_stream\_data = None  graph = []    for vel in VELOCITIES:  message\_flow = Message.get\_list\_of\_messages\_from\_excel()  stream = Stream(flow=message\_flow, velocity=vel, is\_priority=True)  stats = calculate\_stats(data=stream.stream\_data\_by\_type)  all\_type\_stats.append(stats[0])  all\_total\_stats.append(stats[1])  if vel == VELOCITIES[-1]:  final\_stream\_data = stream.get\_stream\_data()  del stream    for i in ["m", "p", 'w', 'u']:    plt.figure('Зависимость ' + i + ' от V') #to let the index start at 1    plt.plot(VELOCITIES, [all\_type\_stats[j][1][i] for j in range(3)], label='Тип 1')  plt.plot(VELOCITIES, [all\_type\_stats[j][2][i] for j in range(3)], label='Тип 2')  plt.plot(VELOCITIES, [all\_type\_stats[j][3][i] for j in range(3)], label='Тип 3')  plt.plot(VELOCITIES, [all\_type\_stats[j][4][i] for j in range(3)], label='Тип 4')  plt.xlabel('V')  plt.ylabel(i)  plt.legend()    for i in ["W", "U", 'L']:    plt.figure('Зависимость ' + i + ' от V') #to let the index start at 1    plt.plot(VELOCITIES, [all\_total\_stats[j][i] for j in range(3)])  plt.plot(VELOCITIES, [all\_total\_stats[j][i] for j in range(3)])  plt.plot(VELOCITIES, [all\_total\_stats[j][i] for j in range(3)])  plt.xlabel('V')  plt.ylabel(i)  plt.show()      print(all\_total\_stats)  to\_excel(type\_stats=all\_type\_stats, total\_stats=all\_total\_stats, stream\_data=final\_stream\_data) |

**messege.py**

|  |
| --- |
| import numpy as np  from config import \*  import pandas as pd  from pandas import read\_excel  previous\_timestamp = None  def get\_time():  global previous\_timestamp  if previous\_timestamp is None:  previous\_timestamp = 0  return previous\_timestamp  res = previous\_timestamp  while res == previous\_timestamp:  res = previous\_timestamp + int(b \* np.random.weibull(a=a))  previous\_timestamp = res  return res  class Message:  ms\_type: int  recipient: int  message\_len: int  timestamp: int  def \_\_init\_\_(self, message\_type: int, recipient: int = None, length: int = None, timestamp: int = None):  if recipient is None or length is None or timestamp is None:  self.ms\_type = message\_type  self.recipient = np.random.choice(RECIPIENTS, p=RECIPIENTS\_PROBABILITIES[self.ms\_type])  self.message\_len = np.random.randint(21, 194)  self.timestamp = get\_time()  else:  self.ms\_type = message\_type  self.message\_len = length  self.recipient = recipient  self.timestamp = timestamp  def \_\_str\_\_(self):  return (f"message\_type: {self.ms\_type}, recipient: {self.recipient}, "  f"length: {self.message\_len}, timestamp: {self.timestamp}")  def dict(self):  return {"message\_type": self.ms\_type, "recipient": self.recipient,  "length": self.message\_len, "timestamp": self.timestamp}  @staticmethod  def get\_list\_of\_messages():  return list(map(lambda ms\_type: Message(ms\_type),  np.random.choice(TYPES, size=MESSAGES\_QUANTITY, p=PROBABILITIES)))  @staticmethod  def get\_list\_of\_messages\_from\_excel(filename: str = "input\_data.xlsx"):  data = read\_excel(filename).to\_dict('records')  return list(map(lambda m: Message(m["message\_type"], m["recipient"], m["length"], m["timestamp"]), data))  if \_\_name\_\_ == '\_\_main\_\_':  messages = sorted(list(map(lambda x: x.dict(), Message.get\_list\_of\_messages())), key=lambda x: x['timestamp'])  df = pd.DataFrame(messages)  excel\_filename = "input\_data.xlsx"  df.to\_excel(excel\_filename, index=False, engine='openpyxl') |

**config.py**

|  |
| --- |
| MESSAGES\_QUANTITY = 100  TYPES = [1, 2, 3, 4]  PROBABILITIES = [0.09, 0.68, 0.04, 0.19]  RECIPIENTS = [1, 2, 3, 4, 5]  RECIPIENTS\_PROBABILITIES = {  1: [0.52, 0.27, 0.09, 0.07, 0.05],  2: [0.34, 0.44, 0.07, 0.12, 0.03],  3: [0.63, 0.11, 0.08, 0.17, 0.01],  4: [0.51, 0.02, 0.23, 0.12, 0.12],  }  a, b = 2, 1.9 # параметры распределения Вейбулла  VELOCITIES = [40, 50, 60] |