

Санкт-Петербургский политехнический университет
Петра Великого

Физико-механический институт

Кафедра «Прикладная математика»

**Отчёт по лабораторной работе
по дисциплине «Компьютерные сети»
Реализация протоколов автоматического запроса повторной
передачи Go-Back-N и Selective Repeat**

Выполнил студент:
Куксенко Кирилл Сергеевич
группа: 5040102/20201

Проверил:
к.ф.-м.н., доцент
Баженов Александр Николаевич

Санкт-Петербург
2023 г.

Содержание

1	Постановка задачи	2
2	Теория	2
3	Реализация	2
4	Результаты	2
5	Обсуждение	9

Список иллюстраций

1	Число сообщений от таймаута (размер окна = 10, вероятность повреждения сообщения = 0.0)	3
2	Время работы от таймаута (размер окна = 10, вероятность повреждения сообщения = 0.0)	3
3	Go-Back-N. Число сообщений от размера окна	4
4	Go-Back-N. Время работы от размера окна	5
5	Selective Repeat. Число сообщений от размера окна	5
6	Selective Repeat. Время работы от размера окна	6
7	Go-Back-N. Число сообщений от вероятности потери сообщения	7
8	Go-Back-N. Время работы от вероятности потери сообщения	7
9	Selective Repeat. Число сообщений от вероятности потери сообщения	8
10	Selective Repeat. Время работы от вероятности потери сообщения	8

1 Постановка задачи

Нужно реализовать систему из двух объектов - отправителя (Sender) и получателя (Receiver), которые будут обмениваться сообщениями по каналу связи с помощью протоколов автоматического запроса повторений передачи с плавающим окном: Go-Back-N и Selective Repeat.

Необходимо выяснить зависимость времени работы и количество посланных сообщений от размера плавающего окна и вероятности потери сообщения для каждого протокола и сравнить друг с другом.

2 Теория

Протоколы Go-Back-N и Selective Repeat являются протоколами скользящего окна. Основное различие между этими двумя протоколами заключается в том, что после обнаружения подозрительного или поврежденного сообщения протокол Go-Back-N повторно передает все сообщения не получившие подтверждения о получении, тогда как протокол Selective Repeat повторно передает только то сообщение, которое оказалось повреждено.

3 Реализация

Весь код написан на языке Python (версии 3.7.3). Для каждого протокола получатель и отправитель работают параллельно в отдельных потоках. [Ссылка на GitHub с исходным кодом.](#)

4 Результаты

Введём две основные метрики, по которым будем сравнивать оба протокола: число сообщений, которые пришлось отправить отправителю, и время работы протокола, за которое получатель смог получить все сообщения без повреждений. Посмотрим на зависимость этих метрик от размера окна, времени таймаута и вероятности потери сообщения.

Во всех тестах (если не сказано обратное) число сообщений, которые получатель должен получить от отправителя равно 100, а таймаут равен 0.5. Замеры времени работы проводились на CPU Intel I7-7700HQ 2.80GHz.

Сначала посмотрим на зависимость числа сообщений и времени работы от размера таймаута. Размер окна равен 10, сообщение не может

быть повреждено рис. 1 и рис. 2.

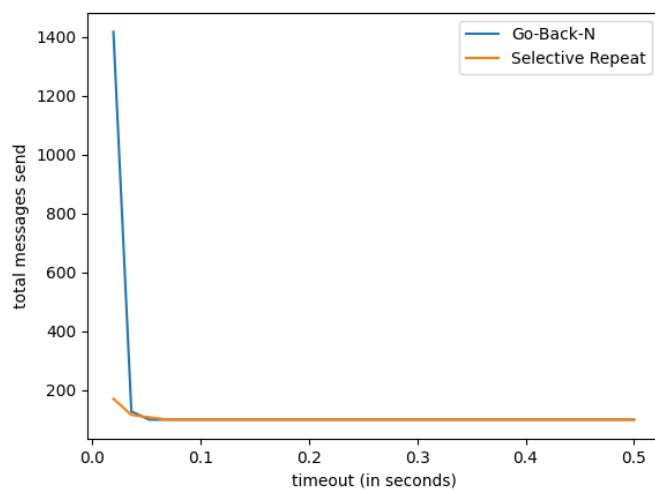


Рис. 1: Число сообщений от таймаута (размер окна = 10, вероятность повреждения сообщения = 0.0)

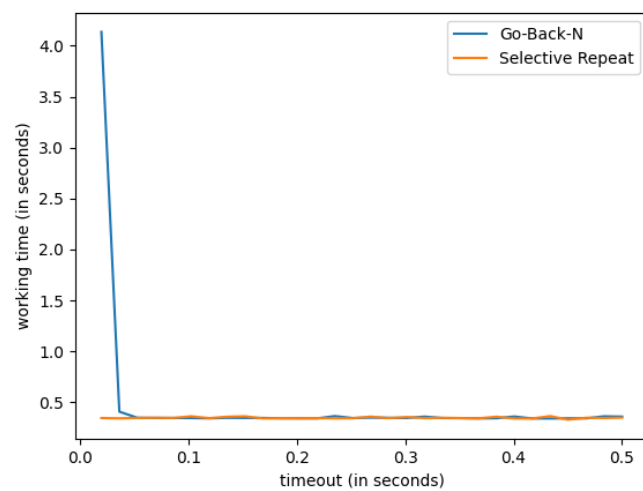


Рис. 2: Время работы от таймаута (размер окна = 10, вероятность повреждения сообщения = 0.0)

Видно, что при очень малых значениях таймаута, отправитель для некоторых сообщений не успевает получить от получателя подтверждения до истечения времени ожидания, из-за чего посылает повторные сообщения. Но с увеличением времени таймаута такие случаи пропадают и число всех отправленных сообщений равно числу успешно переданных сообщений.

Теперь рассмотрим значения числа всех отправленных сообщений и времени работы в зависимости от размера окна и вероятности ошибки. Сначала для протокола Go-Back-N.

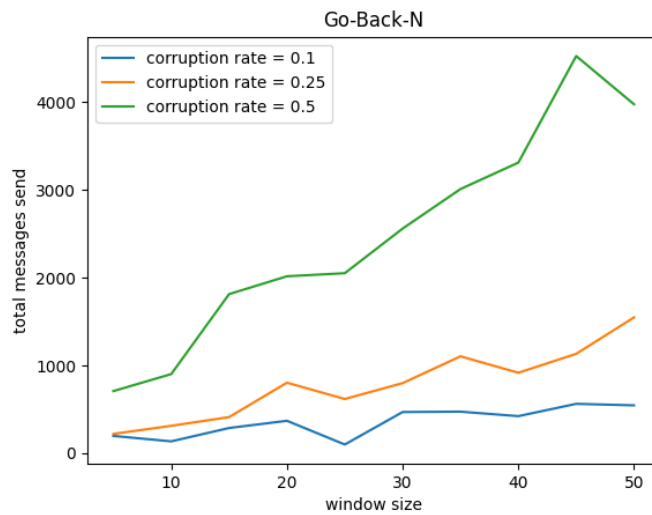


Рис. 3: Go-Back-N. Число сообщений от размера окна

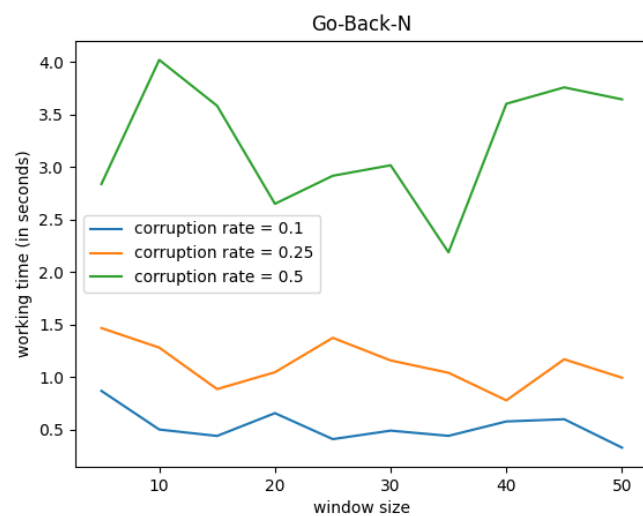


Рис. 4: Go-Back-N. Время работы от размера окна

Затем для протокола Selective Repeat.

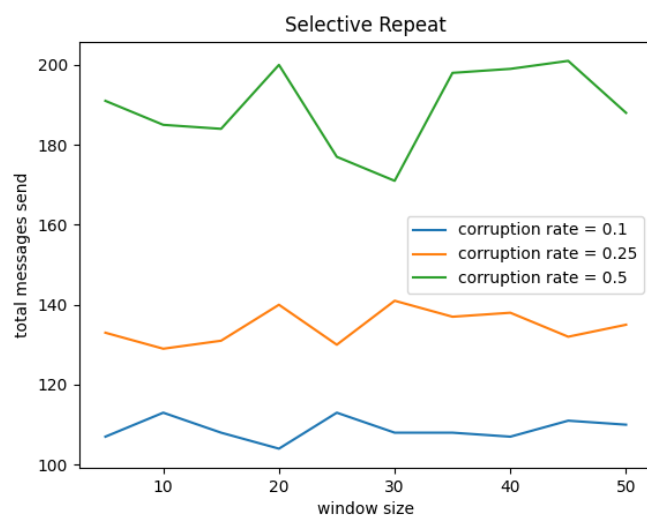


Рис. 5: Selective Repeat. Число сообщений от размера окна

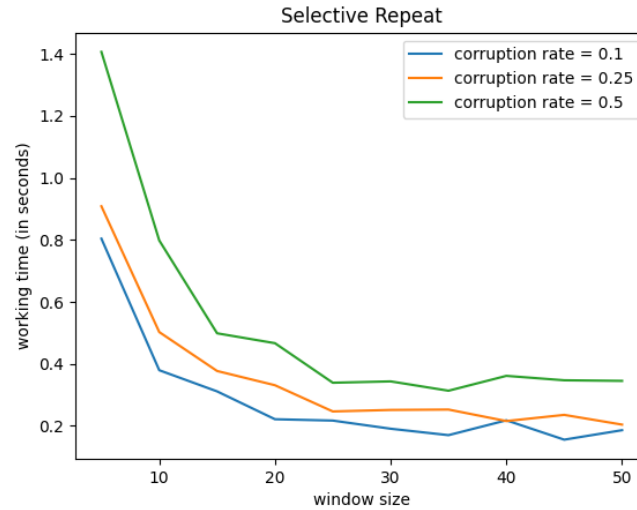


Рис. 6: Selective Repeat. Время работы от размера окна

Как видно из рисунка рис. 3 общее число отправленных сообщений в протоколе Go-Back-N прямо пропорционально размеру окна, что особенно заметно при больших значениях вероятности потери сообщения. Но при этом время работы Go-Back-N не зависит от размера окна рис. 4. С другой стороны, для протокола Selective Repeat размер окна не влияет на общее число отправленных сообщений рис. 5. При этом время работы протокола Selective Repeat тем меньше, чем больше размер окна, но с увеличением размера прирост скорости работы становится всё меньше рис. 6.

Также рассмотрим зависимости тех же метрик от вероятности потери сообщения. Для Go-Back-N имеем.

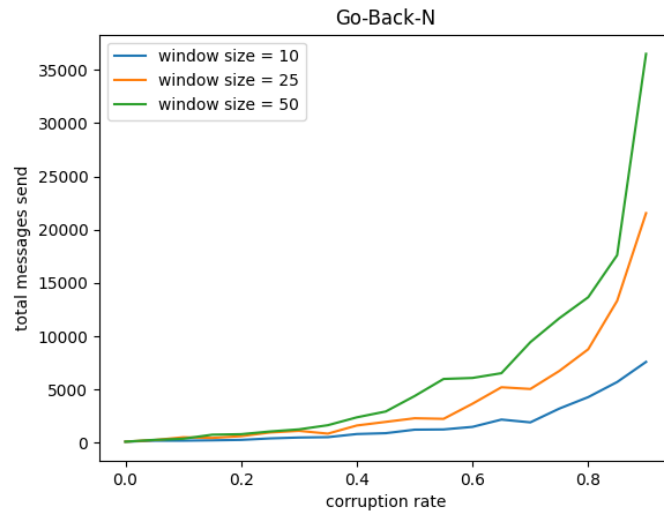


Рис. 7: Go-Back-N. Число сообщений от вероятности потери сообщения

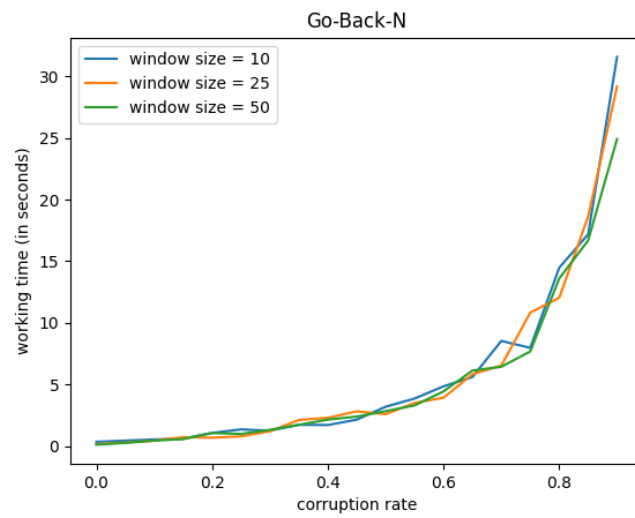


Рис. 8: Go-Back-N. Время работы от вероятности потери сообщения

А для Selective Repeat.

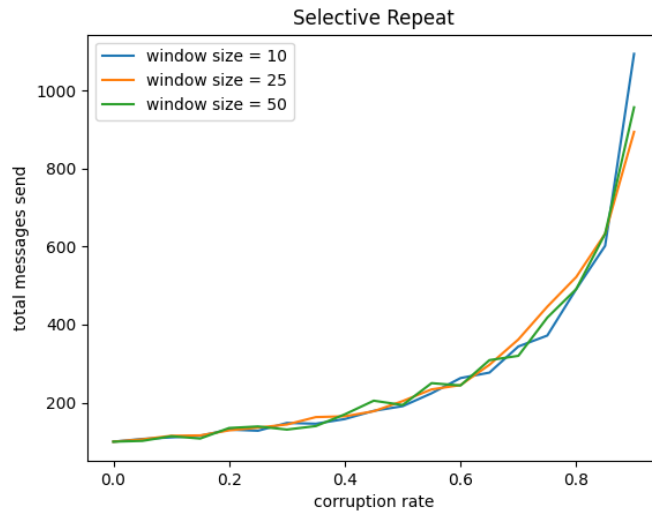


Рис. 9: Selective Repeat. Число сообщений от вероятности потери сообщения

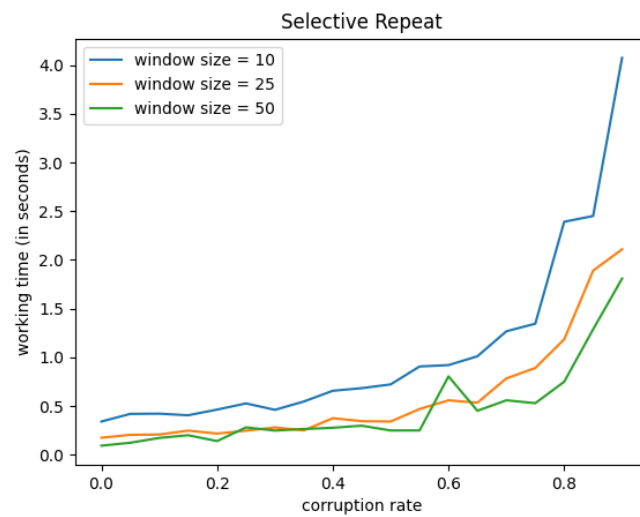


Рис. 10: Selective Repeat. Время работы от вероятности потери сообщения

Как видно на рисунках 7, 9 общее число отправленных сообщений с ростом вероятности потери сообщения у протокола Go-Back-N силь-

но больше, чем у протокола Selective Repeat. Как следствие, Go-Back-N работает значительно дольше, чем Selective Repeat рис. 8, 10.

5 Обсуждение

Из приведённых результатов можно заметить, что в одинаковых условиях протоколу Selective Repeat требуется отправить меньше сообщений, чем протоколу Go-Back-N. Что ожидаемо, в силу разной обработки и повторной передачи потерянных сообщений (раздел 2). Как следствие, протокол Selective Repeat работает значительно быстрее протокола Go-Back-N.