

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА
ВЕЛИКОГО

ФИЗИКО-МЕХАНИЧЕСКИЙ ИНСТИТУТ

ВЫСШАЯ ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ
ФИЗИКИ

Компьютерные сети

Отчёт по лабораторной работе №1

**“Реализация протоколов автоматического запроса
повторной передачи Go-Back-N и Selective Repeat”**

Выполнил:

Студент: Парусов Владимир

Группа: 5040102/30201

Принял:

к. ф.-м. н., доцент

Баженов Александр Николаевич

2024 г.

Contents

1. Постановка задачи	2
2. Теория	2
2.1. Протокол Go-Back-N	3
2.2. Протокол Selective Repeat	3
3. Реализация	4
4. Результаты	5
5. Обсуждение	7
6. Приложения	7

List of Figures

1. Визуализация передачи данных с использованием скользящего окна . . .	2
2. Диаграмма работы протокола Go-Back-N	3
3. Диаграмма работы протокола Selective Repeat	4
4. Визуализация передачи данных с использованием скользящего окна . . .	5
5. Зависимость коэффициента эффективности и времени передачи от размера окна при $p = 0.3$	6

1. Постановка задачи

Необходимо реализовать систему, состоящую из отправителя (Sender) и получателя (Receiver), способных обмениваться сообщениями по каналу связи через протоколы автоматического запроса повторной передачи Go-Back-N (GBN) и Selective Repeat (SRP). Канал связи может допускать потерю пакетов с заданной вероятностью. Требуется добавить возможность выбора размера скользящего окна. Сравнить эффективность работы данных протоколов для разных вероятностей ошибок при передаче данных.

2. Теория

Рассмотрим два компьютера, соединённых проводом. При такой связи необходимо доставлять биты в том же порядке, в котором они были отправлены передающей машиной. Для реализации подобной связи были созданы различные протоколы передачи данных. Часть таких протоколов в своей реализации использует разбиение сегментов данных на следующие 4 вида:

- сегменты, которые были отосланы и имеют подтверждение от приемника
- сегменты, которые были отосланы и не имеют подтверждения от приемника
- сегменты, которые могут быть отосланы
- сегменты, которые не могут быть отосланы

Передавать сегменты всех типов в правильном порядке проблематично из-за возможных ошибок при передаче. Чтобы решить эту проблему, необходимо воспользоваться некоторым окном фиксированного размера, в рамках которого будет происходить доставка сегментов. Визуализация такого подхода представлена на рис. 1.

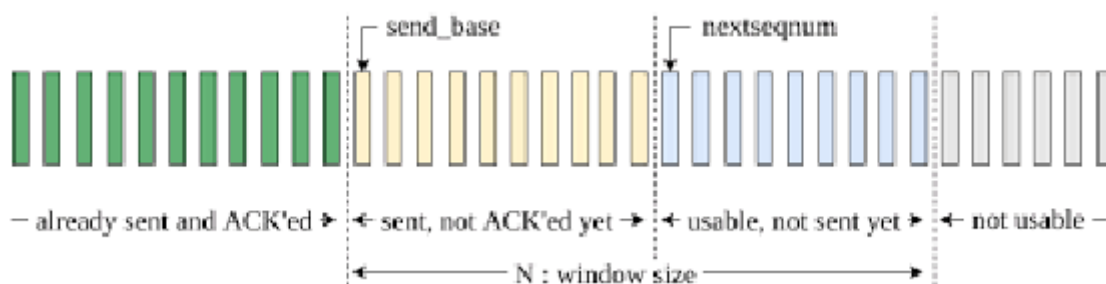


Figure 1. Визуализация передачи данных с использованием скользящего окна

Идея заключается в том, чтобы выявлять и исправлять все ошибки передачи данных в рамках окна. После этого происходит его смещение к сегментам с большим порядковым номером и процедура передачи повторяется. Рассмотрим два протокола, использующих данный подход.

2.1. Протокол Go-Back-N

Особенностью протокола Go-Back-N является отправка всего набора сегментов, находящихся в рамках скользящего окна, не дожидаясь ответа от приемника. Таким образом, после заполнения окна отправленными, но не подтвержденными сегментами, источник ожидает получения подтверждения для всех сегментов. В случае, если один из сегментов не получил подтверждения доставки за некоторое фиксированное время, называемое также таймером, то источник повторяет отправку всех сегментов окна, начиная с этого сегмента. Рассмотрим данный протокол на примере случая, когда размер окна равен четырем (рис. 2).

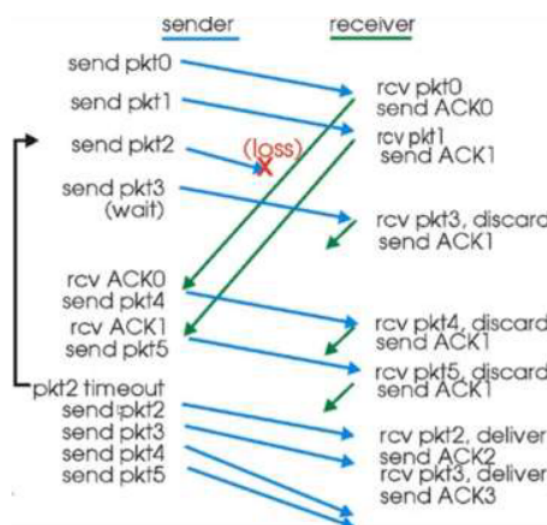


Figure 2. Диаграмма работы протокола Go-Back-N

Источник начинает осуществлять посылку сегментов приемнику. Так как размер окна равен четырем, источник может послать четыре сегмента, т.е. сегменты с номерами 0, 1, 2, 3, без получения подтверждения, после чего источник должен ожидать подтверждения. В указанном примере сегмент номер 2 был потерян при передаче. В связи с этим сегменты 3, 4, 5 поступили вне очереди, поэтому они не должны быть подтверждены приемником. По истечении срока ожидания подтверждения отправитель заново посылает приёмнику весь пул сегментов.

2.2. Протокол Selective Repeat

Протокол Go-Back-N затрачивает довольно много избыточных ресурсов, посылая подтверждённые данные по несколько раз в случае ошибок. В связи с этим, при больших размерах окна и низкой пропускной способности канала передачи становится ярко выраженной низкая эффективность данного протокола. Протокол Selective Repeat позволяет избегать повторной передачи тех сегментов, которые

безошибочно, но вне очереди, были приняты приемником. Повторно передаются только сегменты, переданные с ошибками.

Таким образом, для подтверждения повторно переданного сегмента приемник должен послать источнику индивидуальное подтверждение, и сегменты, пришедшие без ошибок, но вне очереди, должны быть подтверждены. Так же, как и в протоколе Go-Back-N, в Selective Repeat окно размера N используется для ограничения количества посланных, но не подтвержденных сегментов. Но в данном протоколе в окне могут находиться посланные и подтвержденные сегменты. Рассмотрим работу протокола для того же примера, когда размер окна равен четырем (рис. 3).

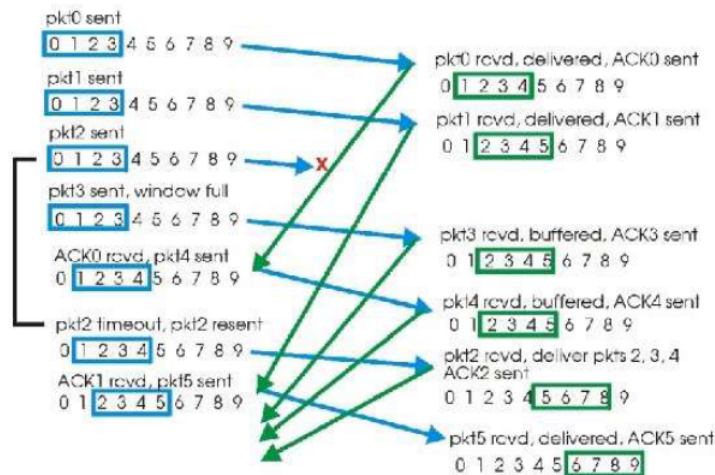


Figure 3. Диаграмма работы протокола Selective Repeat

Как можно заметить, данный протокол является более выгодным при наличии ошибок передачи с точки зрения объема пересылаемых данных, однако сдвиг окна выполняется в соответствии с более сложной логикой: данные должны быть переданы в заданном порядке, поэтому при сдвиге окна нельзя допускать перекрытия старого окна новым. Для этого размер окна не должен превышать половины от количества порядковых номеров.

3. Реализация

Протоколы и эмуляторы исполнителей реализованы в двух отдельных потоках на языке Python. Обмен данными осуществляется через очередь сообщений. Программа состоит из следующих элементов:

- Sender – отправитель, формирует сообщения с данными
- Receiver – получатель, получает сообщения и сообщает о факте доставки
- MsgQueue – канал коммуникации, который хранит сообщения между отправкой и получением, а также имитирует их потерю

Каждый пакет содержит информацию о своем порядковом номере в окне, уникальный номер блока, а также свой статус (доставлен, потерян). Система принимает следующие параметры:

- `protocol` – протокол связи
- `window_size` – величина окна в выбранном протоколе
- `timeout` – время в секундах, после которого пакет считается утерянным в случае отсутствия подтверждения его доставки
- `loss_probability` – вероятность потери сообщения при передаче $[0, 1]$

4. Результаты

Оценку эффективности протоколов будем проводить по двум параметрам:

- коэффициент эффективности k – количество переданных пакетов / количество всех пакетов
- время от начала до конца передачи в секундах – t

Для оценки эффективности была проведена серия экспериментов с различными значениями размера окна и вероятности потери пакетов. Во всех тестах количество передаваемых пакетов равно 100, `timeout` = 0.2 с.

Зависимость коэффициента эффективности k и времени передачи t от вероятности потери пакета p при фиксированном размере окна `window_size` = 3 представлена в таблице 1 и графически на рис. 4.

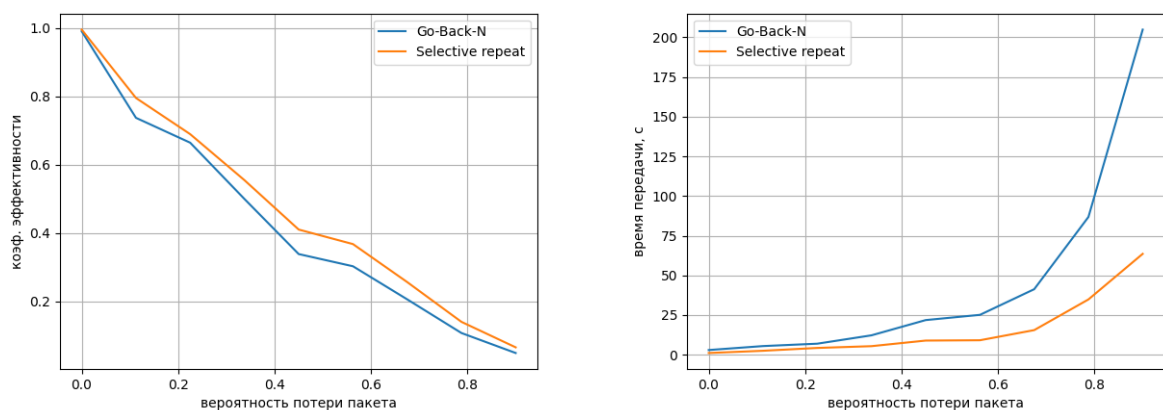


Figure 4. Визуализация передачи данных с использованием скользящего окна

	Go-Back-N		Selective Repeat	
p	k	t	k	t
0.0	2.96	0.99	1.08	1.00
0.1	5.45	0.74	2.49	0.80
0.2	6.95	0.66	4.27	0.69
0.3	12.22	0.50	5.39	0.55
0.5	21.83	0.34	8.92	0.41
0.6	25.13	0.30	9.12	0.37
0.7	41.29	0.21	15.51	0.26
0.8	86.82	0.11	34.85	0.14
0.9	204.89	0.05	63.58	0.07

Table 1. Зависимость эффективности протоколов от вероятности потери пакета при $\text{window_size} = 3$

Зависимость эффективности k и времени передачи t от размера окна window_size при заданной вероятности потери пакета $p = 0.3$ представлена в табл. 2 и на рис. 5.

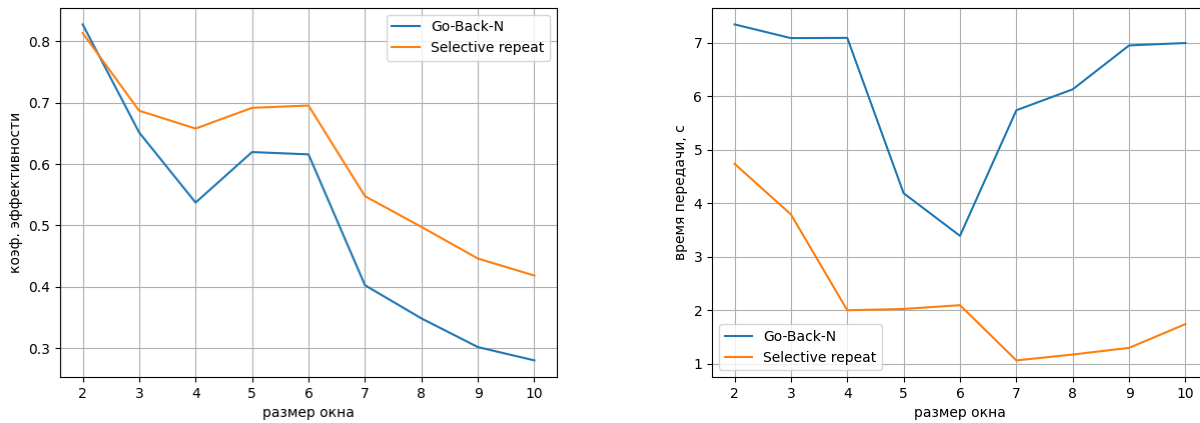


Figure 5. Зависимость коэффициента эффективности и времени передачи от размера окна при $p = 0.3$

	Go-Back-N		Selective Repeat	
window size	k	t	k	t
2	7.34	0.83	4.74	0.81
3	7.09	0.65	3.79	0.69
4	7.09	0.54	2.00	0.66
5	4.19	0.62	2.02	0.69
6	3.39	0.62	2.09	0.70
7	5.74	0.40	1.06	0.55
8	6.13	0.35	1.17	0.50
9	6.95	0.30	1.29	0.45
10	7.00	0.28	1.74	0.42

Table 2. Зависимость коэффициента эффективности и времени передачи от размера окна при $p = 0.3$

5. Обсуждение

Можно заметить, что при малых значениях p (вероятность потери пакета) время передачи сообщений практически не отличается между рассмотренными протоколами. Однако, с увеличением p , протокол Selective Repeat оказывается более эффективным, хотя k (коэффициент эффективности) отличается не более, чем на 10%.

При рассмотрении коэффициента эффективности в зависимости от размера окна, можно заметить что увеличение размера окна приводит к снижению эффективности. Зависимость времени передачи от размера окна для протокола Selective Repeat выглядит практически константной, а для протокола Go-Back-N имеет явно выраженный минимум в значении равном 6.

В целом, из всех рассмотренных зависимостей можно обнаружить что протокол Selective Repeat выдает более эффективный результат, по сравнению с протоколом Go-Back-N.

6. Приложения

1. Репозиторий с кодом программы

<https://github.com/sairsey/compnet>