

Министерство науки и высшего образования Российской
Федерации
Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа прикладной математики и вычислительной физики

Дисциплина «Компьютерные сети»
Отчет по лабораторной работе 1
«Реализация протоколов автоматического запроса повторной передачи
Go-Back-N и Selective Repeat»

Выполнил

Студент группы 5040103/90301

А. А. Северюхина

Принял

к. ф.-м. н., доцент

А. Н. Баженов

Санкт-Петербург

2024

Содержание

1. Постановка задачи.....	3
2. Теория.....	4
Протокол Go-Back-N.....	5
Протокол Selective Repeat.....	6
3. Реализация.....	8
4. Результаты	9
5. Выводы	13
Приложения	14

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 систему из двух компьютеров, соединенных проводом. Данные, которые передаются между устройствами, проходят транспортный уровень взаимодействия в виде одного большого потока байт. Этот поток делится на отдельные части – сегменты, каждый из которых отправляется получателю отдельно. Получатель принимает сегменты и собирает их в один большой поток байт.

Сегменты делят на следующие типы:

1. Могут быть отправлены:
 - Сегменты, которые были отправлены и получили подтверждение от приемника
 - Сегменты, которые были отправлены, но не получили подтверждения от приемника
2. Еще не могут быть отправлены

Для того, чтобы обеспечить гарантию доставки данных, используется подтверждение получения сообщения. Отправитель передает некоторый сегмент данных, получатель принимает сегмент и посылает отправителю подтверждение. Далее отправляется второй сегмент и т.д.

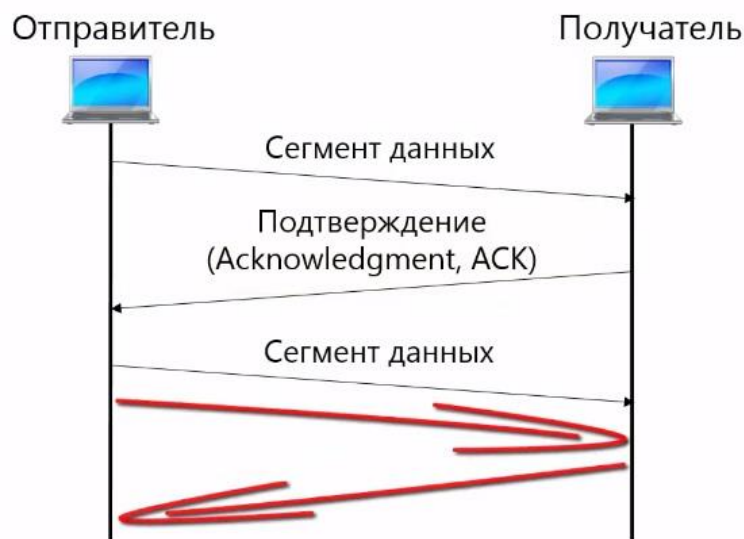


Рисунок 1. Визуализация передачи данных

Когда отправитель посылает сегмент, он устанавливает таймер – время ожидания подтверждения. Если сегмент данных был потерян в сети и не дошел до получателя, то сообщение подтверждения не отправляется и, когда срабатывает таймер, сегмент отправляется повторно.



Рисунок 2. Визуализация потери сегмента данных при передаче информации

При передаче информации с одного устройства на другое необходимо доставлять биты в том же порядке, в котором они были отправлены. Для решения этой проблемы были реализованы методы скользящего окна.

Основное свойство таких протоколов: отправка сегментов происходит в рамках окна фиксированного размера. Данные делятся на части для того, чтобы выделять и исправлять ошибки внутри окна. После этого происходит смещение к следующим сегментам, и процедура повторяется.

Визуализация такого подхода представлена на рис.

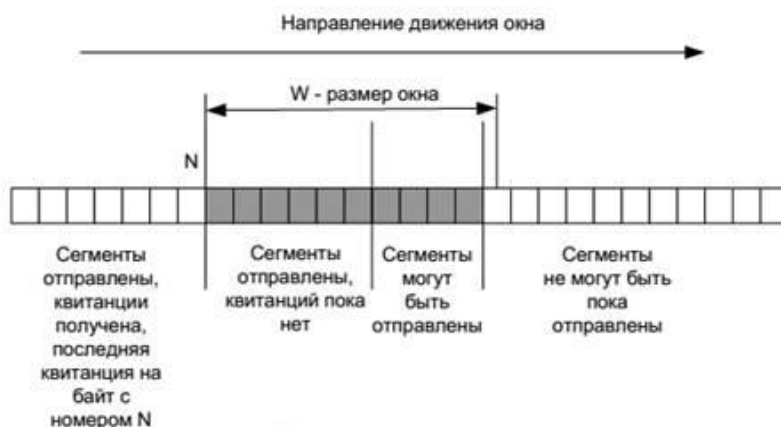


Рисунок 3. Визуализация передачи данных с использованием скользящего окна

Рассмотрим два протокола, использующих данный подход.

Протокол Go-Back-N

Его особенность – отправка всего набора сегментов, находящихся в рамках окна, не дожидаясь ответа приемника. После этого источник ожидает

подтверждения доставки за определенное фиксированное время (таймер). Если источник не получил подтверждения, то он повторяет отправку всех сегментов окна, начиная с того самого сегмента.

Рассмотрим на примере:

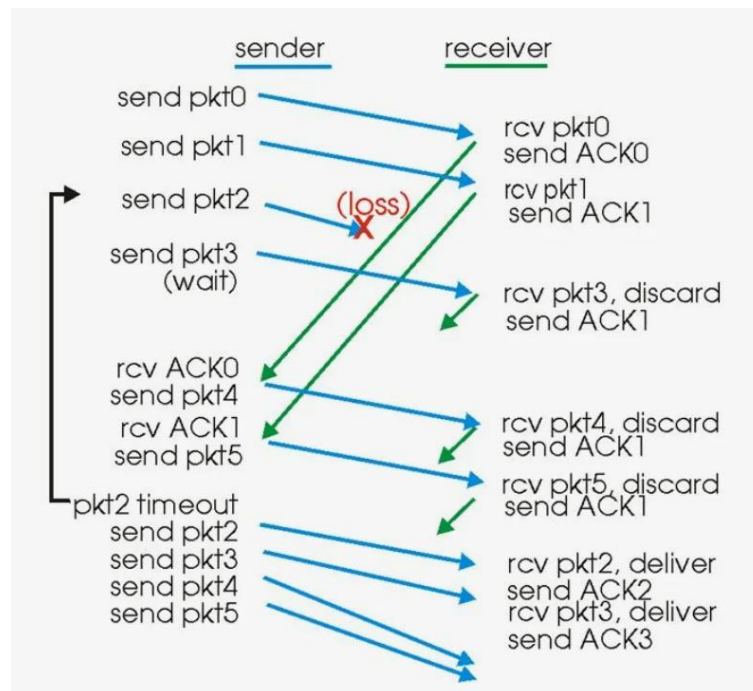


Рисунок 4. Визуализация работы алгоритма Go-Back-N

Источник отправляет сегменты приемнику. Размер скользящего окна в данном примере равен 4. Поэтому на первой итерации мы можем отправить сегменты с индексами 0, 1, 2, 3. Sender получил сообщение с подтверждением АСК (Acknowledge) с 0 и 1 сегмента. Сегмент 2 был потерян при передаче. Так как сообщение от сегмента два не поступило, 3 сегмент был отправлен «вне очереди», и он не может быть подтвержден приемником. После того, как сработает таймер, отправитель заново посылает пул сегментов, начиная уже со 2.

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.

Данный протокол позволяет избегать повторной передачи тех сегментов, которые были приняты. Повторно передаются только те сегменты, которые были переданы с ошибками.

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 отправленных, но не подтвержденных сегментов. Но в данном протоколе в окне могут находиться и отправленные и подтвержденные сегменты.

Рассмотрим работу протокола на примере.

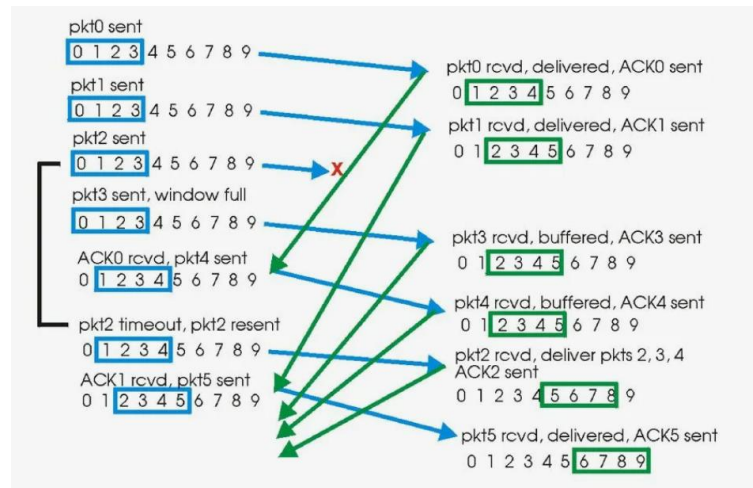


Рисунок 5. Визуализация работы протокола Selective Repeat

При отправке сегмента 2 возникла ошибка, при этом следующие пакеты были отправлены и сообщение об их получении было сохранено в области памяти. После того, как сработал таймер, сегмент был отправлен повторно. Но в отличие от Go-Back-N, получение сегмента 3, пришедшего вне очереди, было подтверждено.

Таким образом, протокол Selective Repeat является более выгодным при наличии ошибок передачи с точки зрения объема пересылаемых данных, но имеет более сложный алгоритм сдвига окна.

3. Реализация

Реализация выполнения протоколов выполнена на языке Python с использованием потоков. Обмен сообщениями происходит в виде очереди сегментов.

Программа содержит следующие глобальные элементы:

- `class MessageStatus` – класс для определения статуса сегмента
- `class Message` – класс со свойствами сообщения
- `class MsgQueue` – класс с методами для работы с сегментами (получение/изменение статуса)
- `GBN_sender` – описывает работу отправителя сообщения для протокола GBN
- `GBN_receiver` - описывает работу получателя сообщения для протокола GBN
- `SRP_sender` - описывает работу отправителя сообщения для протокола SRP
- `SRP_receiver` - описывает работу получателя сообщения для протокола SRP
- `test_different_loss_prob` – функция для тестирования работы протоколов при разных значениях параметра вероятности потери сегмента
- `test_diff_wind_size` - функция для тестирования работы протоколов при разных значениях параметра скользящего окна

Каждое сообщение делится на сегменты. Сегменты имеют свой порядковый номер в окне, уникальный номер блока и статус. Также для работы алгоритма необходимо задать следующие параметры: протокол связи, размер скользящего окна, таймер (по истечении этого времени сегмент считается утерянным), вероятность потери сегмента при передаче.

4. Результаты

Для сравнения работы протоколов передачи связи были проведены вычисления коэффициента эффективности и времени передачи сообщения.

Коэффициент эффективности = количества переданных сегментов / общее количество сегментов.

Было проведено две серии экспериментов с различными значениями параметров. Первый - с изменением значения параметра вероятности потери сегмента в диапазоне $[0, 0.9]$ с шагом 0.1. Второй – с изменением параметра размера скользящего окна в диапазоне $[2, 10]$ с шагом 1.

Зависимость коэффициента эффективности и времени передачи сообщения от вероятности потери сегмента при фиксированном размере окна $window_size = 4$ представлена на рисунках 6, 7 и в таблице на рисунке 8.

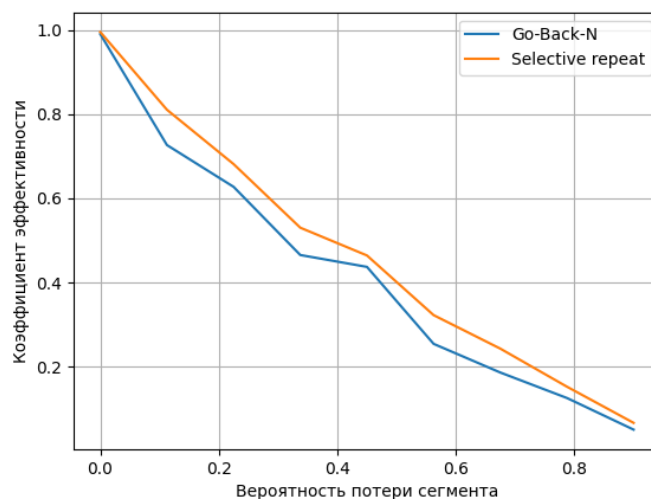


Рисунок 6. Визуализация зависимости коэффициента эффективности от значения вероятности потери сегмента

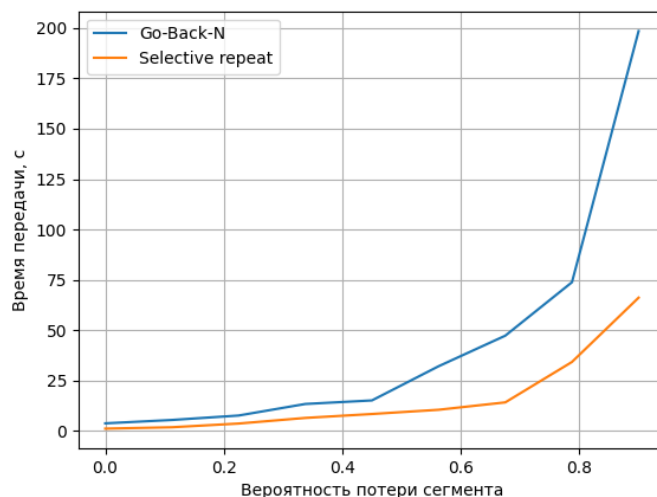


Рисунок 7. Визуализация зависимости времени передачи от значения вероятности потери сегмента

```

-----
test with different lossing propability
-----

```

p	GBN		SRP	
	t	k	t	k
0.0	3.05	0.99	1.10	1.00
0.1	4.71	0.77	2.64	0.83
0.2	8.46	0.60	3.80	0.68
0.3	12.99	0.47	5.28	0.53
0.5	18.97	0.37	6.11	0.47
0.6	32.09	0.25	10.38	0.33
0.7	41.58	0.21	16.36	0.26
0.8	82.13	0.12	31.31	0.15
0.9	208.56	0.05	60.88	0.07

Рисунок 8. Таблица значений времени передачи сообщения и эффективности при различных значениях вероятности потери сегмента

Зависимость коэффициента эффективности и времени передачи сообщения от размера скользящего окна при фиксированной вероятности потери сегмента $\text{loss_probability} = 0.3$ представлена на рисунках 9, 10 и в таблице на рисунке 11.

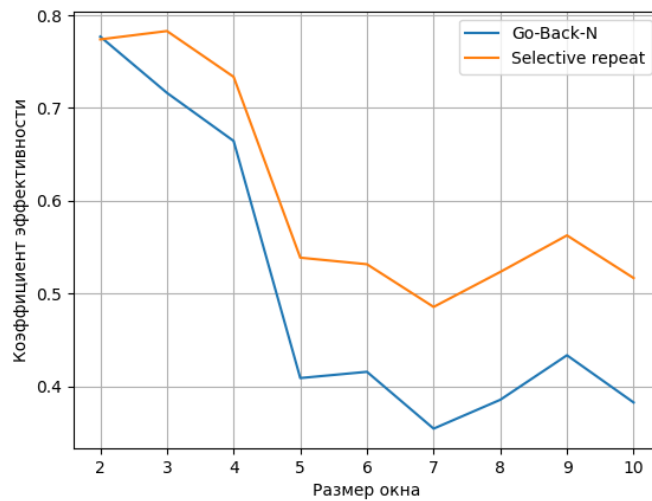


Рисунок 9. Визуализация зависимости коэффициента эффективности от размера скользящего окна

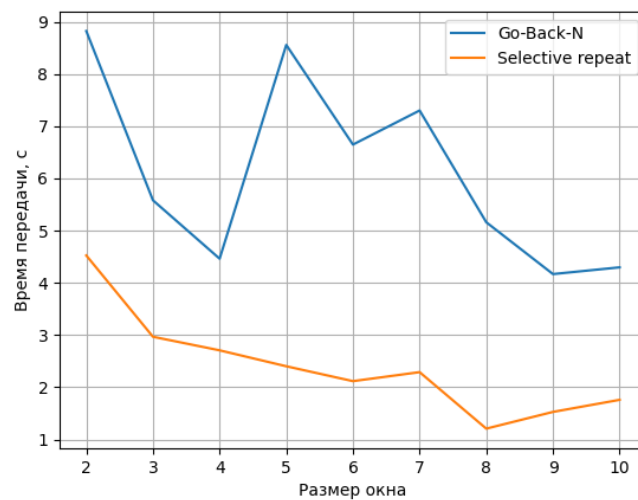


Рисунок 10. Визуализация зависимости времени передачи сообщения от размера скользящего окна

test with different window probability				
w	GBN		SRP	
	t	k	t	k
2	8.82	0.78	4.52	0.77
3	5.58	0.72	2.97	0.78
4	4.46	0.66	2.71	0.73
5	8.56	0.41	2.40	0.54
6	6.65	0.42	2.12	0.53
7	7.30	0.35	2.29	0.49
8	5.16	0.39	1.21	0.52
9	4.17	0.43	1.53	0.56
10	4.30	0.38	1.76	0.52

Рисунок 11. Таблица значений времени передачи сообщения и эффективности при различных размерах скользящего окна

5. Выводы

Из графиков зависимости коэффициента эффективности передачи сообщения можно заметить, что увеличение размера скользящего окна приводит к снижению эффективности работы протоколов.

Из графиков зависимости времени передачи сообщения можно заметить, что при малых значениях вероятности потери сегмента время передачи сообщения практически не отличается для разных протоколов. Но с увеличением `loss_probability`, протокол Selective Repeat оказывается эффективнее.

По результатам полученных зависимостей можно сделать вывод о том, что протокол Selective Repeat выполняет передачу сообщений с вероятностью потери сегментов эффективнее, чем протокол Go-Back-N.

Приложения

1. Репозиторий, содержащий программу реализации передачи сообщений и отчет

<https://github.com/AnastasyaSeveryukhina/interval-and-networks>