Санкт-Петербургский политехнический университет Петра Великого

Физико-механический институт

# Компьютерные сети

## Отчёт по лабораторной работе №1

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

Выполнил:

Студент: Завьялов Валерий

Группа: 5040102/30201

Принял:

к. ф.-м. н., доцент

Баженов Александр Николаевич

2024 г.

# СОДЕРЖАНИЕ

## 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

Представим два компьютера, соединённых проводом. При такой связи важно, чтобы биты доставлялись в том же порядке, в каком они были отправлены с передающей машины. Для этого были разработаны различные протоколы передачи данных. Часть таких протоколов в своей реализации использует разбиение сегментов данных на следующие 4 вида:

- сегменты, которые уже отправлены и получили подтверждение от получателя
- сегменты, которые отправлены, но подтверждение ещё не получено
- сегменты, которые могут быть отправлены
- сегменты, которые пока отправить нельзя

Передача всех этих сегментов в правильном порядке может быть затруднена из-за возможных ошибок. Для решения этой проблемы используется окно фиксированного размера, в пределах которого происходит передача сегментов. Пример такого подхода показан на рис. 1.

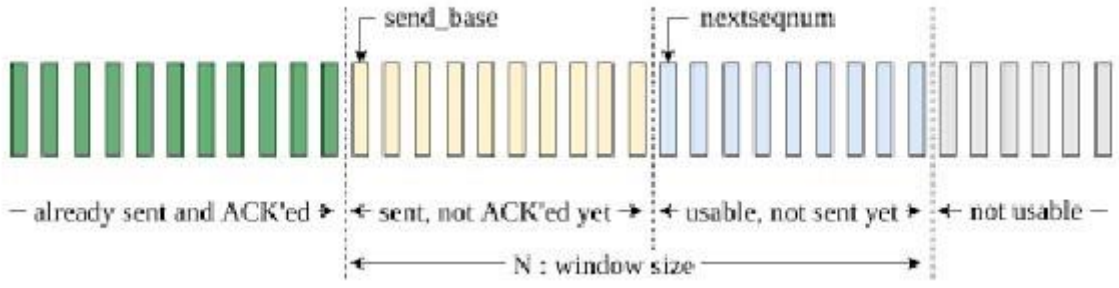



Рис. 1. Визуализация передачи данных с применением скользящего окна

Суть метода заключается в обнаружении и исправлении всех ошибок передачи данных в пределах окна. После этого окно сдвигается к сегментам с более высокими порядковыми номерами, и процесс передачи повторяется. Рассмотрим два протокола, использующих этот подход.

## 2.1 Протокол Go-Back-N

Особенность протокола Go-Back-N заключается в том, что источник отправляет все сегменты внутри скользящего окна, не ожидая подтверждения от получателя. После того как окно заполнено отправленными, но неподтверждёнными сегментами, отправитель ждёт подтверждений для каждого из них. Если один из сегментов не получил подтверждения в течение фиксированного времени, называемого таймером, отправитель повторно отправляет все сегменты, начиная с этого. Рассмотрим работу протокола на примере, где размер окна равен четырём (рис. 2).

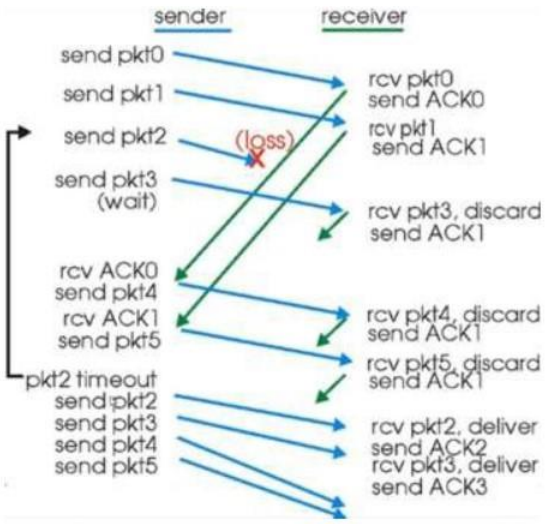



Рис. 2. Диаграмма работы протокола Go-Back-N

Источник начинает отправлять сегменты получателю. Поскольку размер окна составляет четыре, он может отправить четыре сегмента подряд — с номерами 0, 1, 2 и 3 — не дожидаясь подтверждений, после чего должен ожидать ответа. В данном примере сегмент с номером 2 был потерян в процессе передачи, в результате чего сегменты 3, 4 и 5 поступили не по порядку и не были подтверждены получателем. По

истечении времени ожидания отправитель повторно отправляет весь набор сегментов.

## 2.2 Протокол Selective Repeat

Протокол Go-Back-N расходует много лишних ресурсов, так как повторно отправляет уже подтверждённые данные при возникновении ошибок. Это делает его неэффективным при больших размерах окна и низкой пропускной способности канала. Протокол Selective Repeat решает эту проблему, избегая повторной отправки сегментов, которые были приняты без ошибок, но не в порядке очереди. В этом случае повторяются только сегменты с ошибками.

Для подтверждения повторно переданного сегмента получатель отправляет отправителю индивидуальное подтверждение, а сегменты, принятые без ошибок, но не по порядку, также должны быть подтверждены. Как и в Go-Back-N, в Selective Repeat используется окно размера N для ограничения числа отправленных, но неподтверждённых сегментов. Однако в этом протоколе в окне могут находиться как отправленные, так и подтверждённые сегменты. Рассмотрим работу этого протокола на том же примере с окном размером четыре (рис. 3).

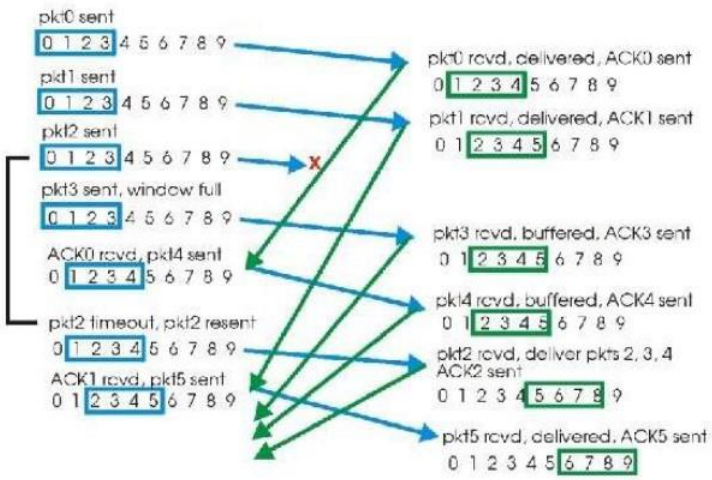



Рис. 3. Диаграмма работы протокола Selective Repeat

Как можно заметить, данный протокол более эффективен в условиях ошибок передачи, поскольку сокращает объём повторно отправляемых данных. Однако сдвиг окна требует более сложной логики: данные должны быть переданы в определённой

последовательности, и при сдвиге окна важно избежать перекрытия старого и нового окна. Чтобы этого достичь, размер окна не должен превышать половину от общего количества порядковых номеров.

## 3. РЕАЛИЗАЦИЯ

Протоколы и эмуляторы исполнителей реализованы на языке Python в двух отдельных потоках, взаимодействующих через очередь сообщений. Программа состоит из следующих компонентов:

- Sender – отправитель, формирующий сообщения с данными,
- Receiver – получатель, принимающий сообщения и подтверждающий их доставку,
- MsgQueue – коммуникационный канал, который хранит сообщения между отправкой и получением, а также симулирует их потерю.

Каждый пакет содержит информацию о своём порядковом номере в окне, уникальный номер блока и статус (доставлен или потерян). Система принимает следующие параметры:

- protocol – используемый протокол связи,
- window_size – размер окна для выбранного протокола,
- timeout – время (в секундах), по истечении которого пакет считается утерянным, если не было получено подтверждение,
- loss_probability – вероятность потери сообщения при передаче (в диапазоне от 0 до 1).

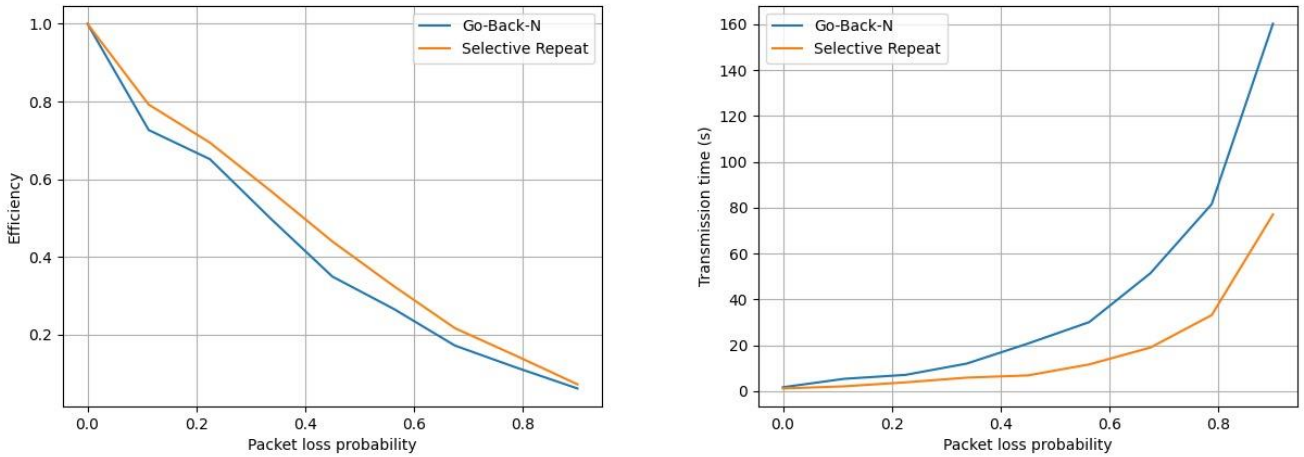
## 4. РЕЗУЛЬТАТЫ

Оценку эффективности протоколов будем проводить по двум параметрам:

- коэффициент эффективности k – количество всех пакетов / количество переданных 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.




Визуализация передачи данных с использованием скользящего окна .4 .Рис

Таблица 1. Зависимость эффективности протоколов от вероятности потери пакета при window_size = 3

| p | Go-Back-N | | Selective Repeat | |
|---|---|---|---|---|
| | k | t | k | t |
| 0.0 | 1.63 | 1.00 | 1.10 | 1.00 |
| 0.1 | 5.31 | 0.73 | 2.03 | 0.79 |
| 0.2 | 7.02 | 0.65 | 3.76 | 0.69 |
| 0.3 | 11.95 | 0.50 | 5.82 | 0.57 |
| 0.5 | 20.70 | 0.35 | 6.80 | 0.44 |
| 0.6 | 30.04 | 0.27 | 11.60 | 0.33 |
| 0.7 | 51.39 | 0.17 | 18.96 | 0.22 |
| 0.8 | 81.52 | 0,12 | 33.10 | 0.15 |
| 0.9 | 160.25 | 0.06 | 77.03 | 0.07 |

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.

\

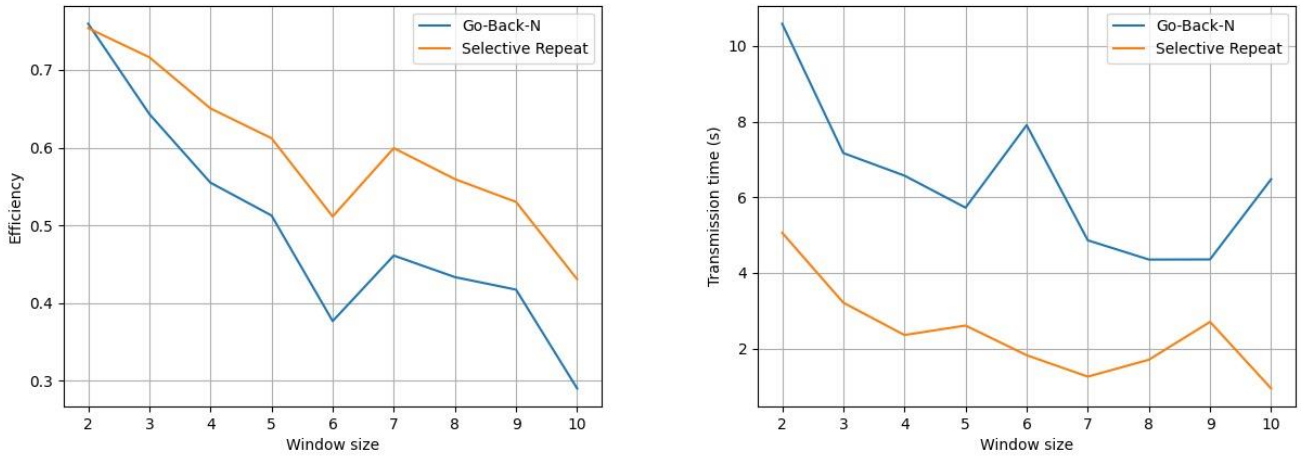



Рис. 5. Зависимость коэффициента эффективности и времени передачи от размера окна при p =0.3

Таблица 2. Зависимость эффективности протоколов от вероятности потери пакета при window_size = 3

| window size | Go-Back-N | | Selective Repeat | |
|---|---|---|---|---|
| | k | t | k | t |
| 2 | 10.59 | 0.76 | 5.06 | 0.75 |
| 3 | 7.17 | 0.64 | 3.21 | 0.72 |
| 4 | 6.57 | 0.55 | 2.35 | 0.65 |
| 5 | 5.72 | 0.51 | 2.61 | 0.61 |
| 6 | 7.91 | 0.38 | 1.82 | 0.51 |
| 7 | 4.86 | 0.46 | 1.25 | 0.60 |
| 8 | 4.35 | 0.43 | 1.70 | 0.56 |
| 9 | 4.35 | 0.42 | 2.70 | 0.53 |
| 10 | 6.48 | 0.29 | 0.94 | 0.43 |

## 5. ОБСУЖДЕНИЕ

Заметим, что при низкой вероятности потери пакетов производительность протоколов (время передачи) практически одинакова. Однако с увеличением вероятности потерь протокол Go-Back-N всё сильнее уступает Selective Repeat в производительности. При этом коэффициент эффективности для обоих протоколов отличается не более чем на 10% при любых значениях p.

Во второй группе поведение коэффициента эффективности в зависимости от размера окна схоже для обоих протоколов: на графиках наблюдается чётко выраженный минимум. Зависимость времени передачи от размера окна для протокола Selective Repeat становится практически постоянной, начиная с окна размером 4, тогда как для Go-Back-N она выглядит хаотичной.

## 6. ПРИЛОЖЕНИЯ

Репозиторий с кодом программы и кодом отчёта: https://github.com/Nochoooo/networks