

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: Чибышев Тимофей
Группа: 5040102/40201

Принял:

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

2025 г.

Содержание

| | |
|--|----|
| 1. Постановка задачи | 2 |
| 2. Теория | 2 |
| 2.1. Протокол Go-Back-N | 3 |
| 2.2. Протокол Selective Repeat | 3 |
| 3. Реализация | 4 |
| 3.1. Архитектура системы | 5 |
| 3.2. Модель канала связи | 5 |
| 3.3. Реализация протоколов | 5 |
| 3.3.1. Go-Back-N (GBN) | 5 |
| 3.3.2. Selective Repeat (SR) | 5 |
| 3.4. Управление событиями | 6 |
| 4. Результаты | 6 |
| 4.1. Экспериментальная установка | 6 |
| 4.2. Анализ результатов | 6 |
| 4.2.1. Зависимость от вероятности потерь | 6 |
| 4.2.2. Зависимость от размера окна | 7 |
| 4.2.3. Ключевые наблюдения | 8 |
| 4.3. Визуализация результатов | 8 |
| 5. Выводы | 9 |
| 6. Приложения | 10 |

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.

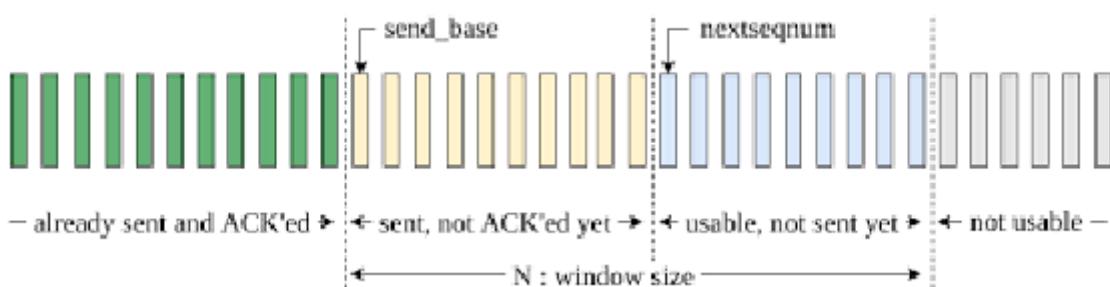


Рис. 1. Визуализация передачи данных с использованием скользящего окна

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).

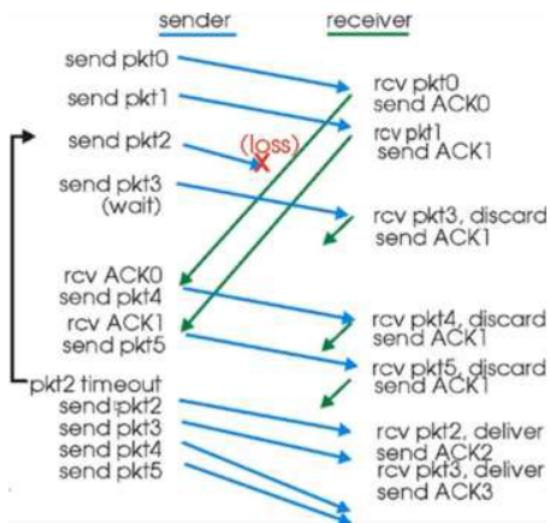


Рис. 2. Диаграмма работы протокола Go-Back-N

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 посыпает приемнику весь пул сегментов.

2.2. Протокол Selective Repeat

Протокол Go-Back-N затрачивает довольно много избыточных ресурсов, посыпая 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).

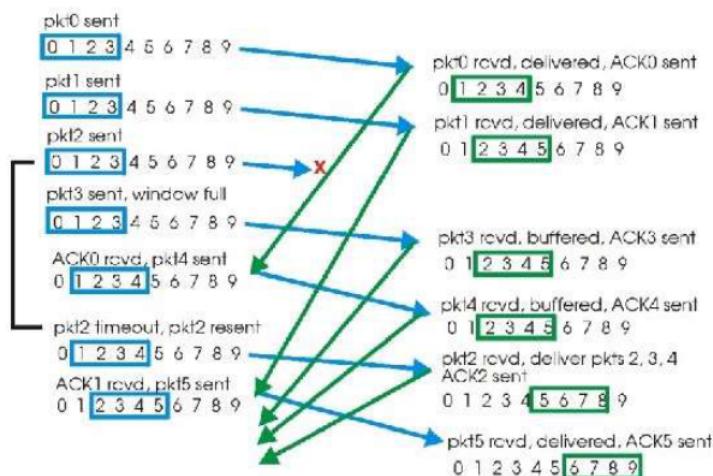


Рис. 3. Диаграмма работы протокола Selective Repeat

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

Для исследования характеристик протоколов GBN и SR была разработана дискретно-событийная модель на языке Python. Реализация включает следующие ключевые компоненты:

3.1. Архитектура системы

Система построена на основе дискретно-событийного подхода, где все взаимодействия моделируются как события в виртуальном времени. Основные структурные элементы:

- **Simulator** – центральный класс, управляющий симуляцией и содержащий состояние протоколов
- **Packet** – модель пакета данных или подтверждения (ACK)
- **Slot** – слот окна для протокола Selective Repeat
- **Event** – событие в системе (отправка, доставка, таймаут)

3.2. Модель канала связи

Канал связи реализован с вероятностной моделью потерь:

- Вероятность потери пакета данных: `loss_probability`
- Вероятность потери ACK: также `loss_probability`
- Фиксированные задержки распространения: `data_delay` и `ack_delay`
- Реализация потерь через случайную генерацию с заданной вероятностью

3.3. Реализация протоколов

3.3.1. Go-Back-N (GBN)

- Использует скользящее окно с базовым номером (`base`) и следующим номером для отправки (`nextseq`)
- Таймер активируется для первого неподтвержденного пакета
- При таймауте повторно передаются все пакеты в окне, начиная с базового
- Получатель принимает только in-order пакеты, отправляя кумулятивные ACK

3.3.2. Selective Repeat (SR)

- Реализован через систему фиксированных слотов (`slots`), каждый с независимым состоянием
- Каждый слот отслеживает свой виртуальный порядковый номер и статус (`BUSY`, `NEED_RETRANSMIT`, `AVAILABLE`)
- Независимые таймеры для каждого слота
- Получатель буферизует out-of-order пакеты и отправляет индивидуальные ACK
- Размер окна ограничен половиной пространства порядковых номеров

3.4. Управление событиями

Система использует приоритетную очередь событий:

- **Типы событий:** `send_next`, `deliver_data`, `deliver_ack`, `timeout`
- События планируются с временными метками и обрабатываются в хронологическом порядке
- Механизм таймаутов реализован через события с отложенным выполнением

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

4.1. Экспериментальная установка

Проводится два основных эксперимента:

1. **Зависимость от размера окна:** фиксированная вероятность потерь 30%, размеры окон от 2 до 48
2. **Зависимость от вероятности потерь:** фиксированный размер окна 15, вероятность потерь от 0% до 80%

Каждая конфигурация тестируется для обоих протоколов с идентичными параметрами канала, что обеспечивает корректное сравнение.

4.2. Анализ результатов

4.2.1. Зависимость от вероятности потерь

Анализ данных эксперимента при фиксированном размере окна 15 показывает следующие закономерности:

- **При нулевых потерях** оба протокола демонстрируют максимальную эффективность: GBN - 1.0, SR - 0.9996, что подтверждает корректность реализации базовых механизмов передачи.
- **При малых потерях (5-15%)** протокол SR значительно превосходит GBN по эффективности. Например, при 10% потерь:
 - SR: эффективность 0.827, время 38.18 мс

- GBN: эффективность 0.389, время 161.85 мс

Разница в 2.1 раза по эффективности объясняется более совершенным механизмом обработки ошибок в SR.

- **При средних потерях (20-40%)** преимущество SR сохраняется. При 30% потерь:

- SR: эффективность 0.508, время 115.74 мс
- GBN: эффективность 0.132, время 625.65 мс

GBN требует в 4.8 раза больше времени для передачи того же объема данных.

- **При высоких потерях (50-80%)** оба протокола деградируют, но SR сохраняет относительное преимущество. При 60% потерь эффективность SR в 3.6 раза выше, чем у GBN.

4.2.2. Зависимость от размера окна

При фиксированной вероятности потерь 30% наблюдаются следующие тенденции:

- **Маленькие окна (2-6):**

- GBN показывает аномально высокую эффективность (0.585 при $W=2$), но очень большое время передачи (3200 мс)
- SR демонстрирует стабильную эффективность около 0.5 при значительно меньшем времени передачи

- **Средние окна (8-20):**

- Эффективность GBN резко падает с 0.227 ($W=8$) до 0.110 ($W=20$)
- Эффективность SR остается стабильной в диапазоне 0.475-0.498
- Время передачи SR монотонно уменьшается с ростом размера окна

- **Большие окна (22-48):**

- GBN продолжает деградировать, эффективность падает до 0.047 при $W=48$
- SR сохраняет эффективность около 0.48-0.49
- Время передачи SR достигает минимума 43.18 мс при $W=48$

4.2.3. Ключевые наблюдения

- **Стабильность SR:** Протокол Selective Repeat демонстрирует стабильную эффективность на всем диапазоне размеров окон (0.475-0.502), в то время как GBN сильно зависит от размера окна.
- **Влияние вероятности потерь:** С ростом вероятности потерь эффективность SR снижается плавно, тогда как GBN демонстрирует резкую деградацию.
- **Время передачи:** SR обеспечивает значительно меньшее время передачи при одинаковых условиях, особенно при высоких вероятностях потерь.

4.3. Визуализация результатов

Визуализация результатов эксперимента по исследованию зависимости коэффициента эффективности и времени передачи пакетов от вероятности потерь пакетов представлена на рис.4.

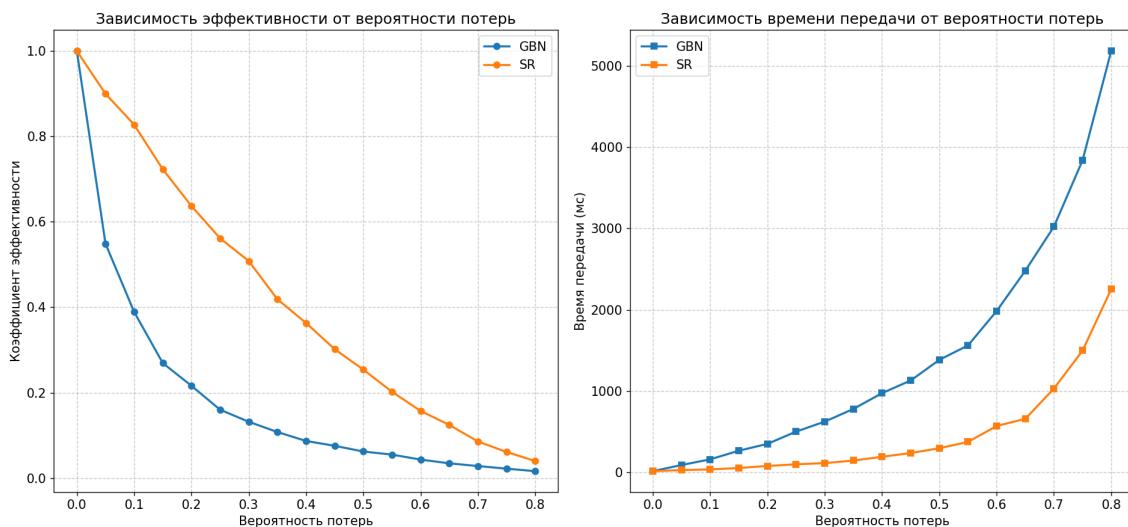


Рис. 4. Сравнение протоколов при разной вероятности потерь пакетов

На рис.5 можем видеть график разности коэффициентов эффективности протоколов SelectiveRepeat и Go-Back-N, что позволяет понять, при каких вероятностях потерь достигается наибольшая разность в коэффициентах эффективности между исследуемыми протоколами.

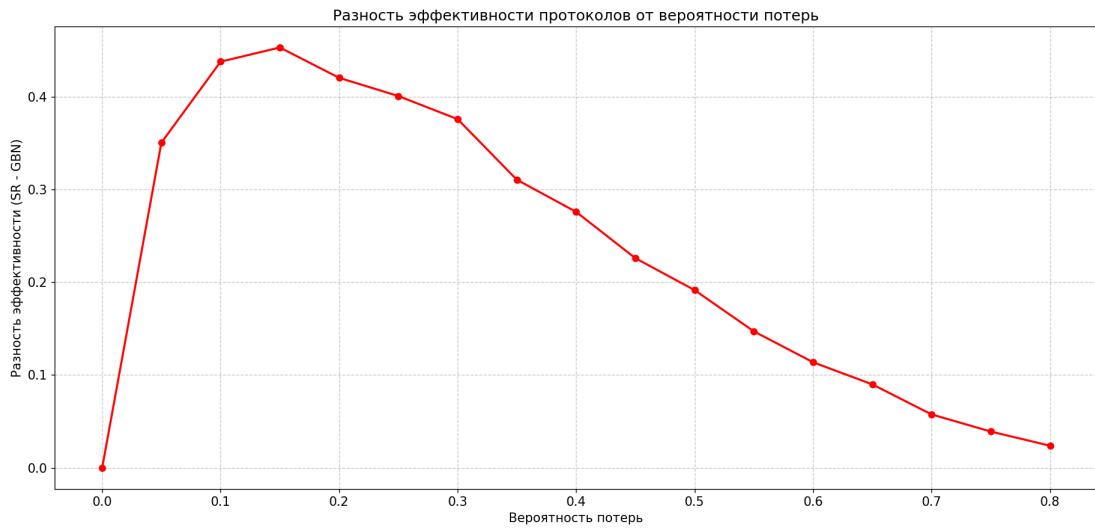


Рис. 5. Разность коэффициентов эффективности протоколов в зависимости от вероятности потери пакетов

Зависимость эффективности k и времени передачи t от размера окна при заданной вероятности потери пакета $p = 0.3$ представлена на рис. 6.

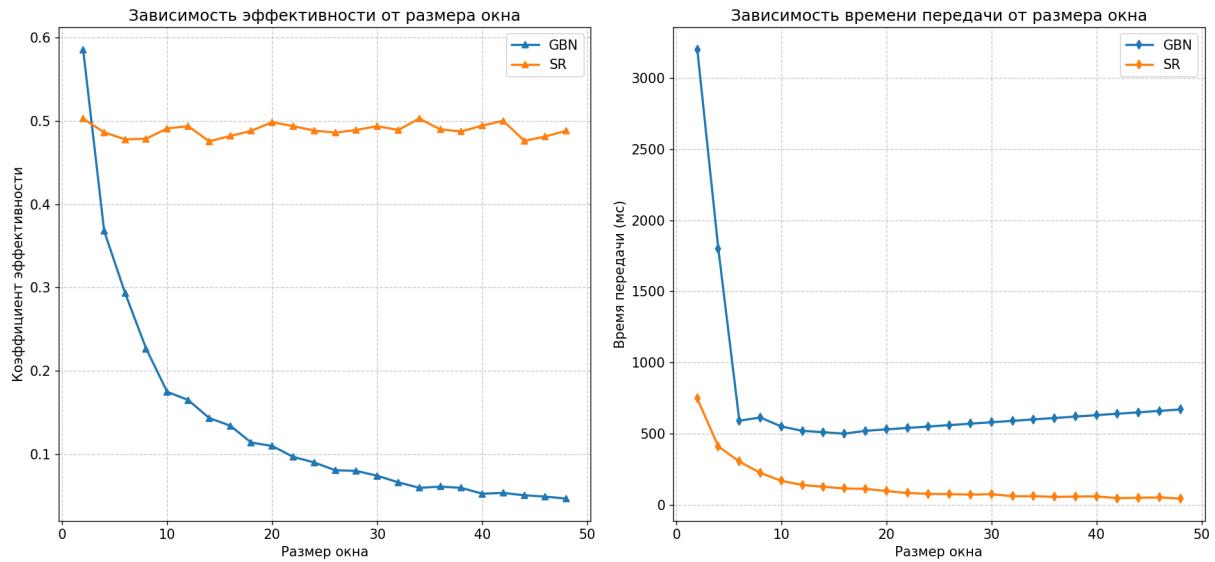


Рис. 6. Зависимость коэффициента эффективности и времени передачи от размера окна при $p = 0.3$

5. Выводы

1. Протокол Selective Repeat демонстрирует значительно лучшую эффективность при всех значениях вероятности потерь и размерах окна.
2. Преимущество SR наиболее выражено при вероятностях потерь 20-40%, где разность эффективности достигает максимума.

3. Go-Back-N сильно деградирует с ростом размера окна в условиях высоких потерь, в то время как SR сохраняет стабильную эффективность.
4. Для практического применения в каналах с потерями более 5% рекомендуется использовать Selective Repeat как более эффективный протокол.
5. Оптимальный размер окна для SR в условиях 30% потерь составляет 20-30 пакетов, что обеспечивает баланс между эффективностью и временем передачи.

6. Приложения

1. Репозиторий с кодом программы и кодом отчёта:

<https://github.com/Timofey-Chibyshev/comppnet>