Санкт-Петербургский политехнический университет имени Петра Великого

Институт прикладной математики и механики

Высшая школа прикладной математики и физики

Отчет
по лабораторной работе #1
по дисциплине
“Компьютерные сети”

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

Выполнил:

Студент: Шварц Александр
Группа: 5040102/40201

Принял:

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

Санкт-Петербург
2026 г.

Содержание

1 Введение

Протоколы автоматического запроса повторной передачи (ARQ) обеспечивают надёжную доставку данных по ненадёжным каналам связи. В данной работе реализованы и сравнены два протокола скользящего окна:

- **Go-Back-N (GBN)** — при потере пакета отправитель повторно передаёт все пакеты, начиная с потерянного;
- **Selective Repeat (SR)** — при потере пакета повторно передаётся только потерянный пакет.

Цель работы — реализовать оба протокола, провести эксперименты с различными параметрами (вероятность потери, размер окна) и сравнить их эффективность.

2 Теоретические основы

2.1 Go-Back-N

В протоколе GBN отправитель может иметь до N неподтверждённых пакетов. Получатель принимает пакеты строго по порядку: если пакет с ожидаемым номером получен, отправляется кумулятивное подтверждение (ACK); если пришёл пакет не по порядку, он отбрасывается.

При тайм-ауте отправитель повторно передаёт *все* пакеты в окне, начиная с базового. Это приводит к значительному количеству повторных передач при высоких вероятностях ошибок.

2.2 Selective Repeat

В протоколе SR получатель буферизует пакеты, пришедшие не по порядку, и отправляет индивидуальные ACK для каждого полученного пакета. При тайм-ауте отправитель повторно передаёт только конкретный неподтверждённый пакет.

Это значительно снижает количество повторных передач по сравнению с GBN, но требует буфера на стороне получателя.

3 Реализация

Система реализована на языке C++ и включает следующие компоненты:

- `Packet` — структура пакета (номер последовательности, данные, флаг ACK);
- `Channel` — канал связи с заданной вероятностью потери;
- `GBN_Sender`, `GBN_Receiver` — отправитель и получатель GBN;
- `SR_Sender`, `SR_Receiver` — отправитель и получатель SR.

Моделирование выполняется пошагово (tick-based): на каждом шаге отправитель посылает пакеты, получатель обрабатывает их и отправляет подтверждения, отправитель принимает ACK и проверяет тайм-ауты.

Допущения модели. Симулятор использует абстрактную модель с мгновенной доставкой (RTT = 0): отправка пакета, его приём получателем и возврат ACK происходят в

рамках одного такта. В такой модели размер окна W выступает как множитель мгновенной пропускной способности (количество пакетов, отправляемых за один такт), а не как классическое сетевое окно, компенсирующее задержку канала. Это упрощение не влияет на сравнение эффективности протоколов (доля полезных передач η), однако абсолютные значения пропускной способности (пакетов/такт) следует интерпретировать с учётом данного допущения.

3.1 Параметры эксперимента

- Количество передаваемых пакетов: 1000
- Тайм-аут: 10 тактов
- Размеры окна: $\{1, 2, 4, 8, 16, 32\}$
- Вероятности потери: $\{0; 0{,}05; 0{,}10; 0{,}20; 0{,}30; 0{,}50\}$
- Количество запусков для усреднения: 5

4 Результаты экспериментов

4.1 Эффективность (доля полезных передач)

Эффективность определяется как отношение числа уникальных пакетов к общему числу отправленных пакетов:

$$\eta = \frac{N_{\text{уник}}}{N_{\text{всего}}}$$

Таблица 1: Эффективность протоколов при различных параметрах

	$p = 0{,}05$		$p = 0{,}10$		$p = 0{,}20$		$p = 0{,}50$	
W	GBN	SR	GBN	SR	GBN	SR	GBN	SR
1	0.901	0.901	0.805	0.805	0.639	0.639	0.255	0.255
2	0.899	0.901	0.796	0.806	0.635	0.639	0.279	0.255
4	0.823	0.901	0.698	0.806	0.499	0.639	0.197	0.255
8	0.700	0.901	0.529	0.805	0.342	0.640	0.111	0.256
16	0.541	0.901	0.368	0.806	0.202	0.639	0.060	0.255
32	0.360	0.901	0.227	0.806	0.109	0.640	0.032	0.258

4.2 Пропускная способность (пакетов за такт)

Таблица 2: Пропускная способность протоколов

	$p = 0{,}05$		$p = 0{,}10$		$p = 0{,}20$		$p = 0{,}50$	
W	GBN	SR	GBN	SR	GBN	SR	GBN	SR
1	0.476	0.476	0.293	0.293	0.151	0.151	0.033	0.033
2	0.949	0.673	0.565	0.395	0.300	0.206	0.075	0.047
4	1.290	0.909	0.770	0.545	0.369	0.288	0.097	0.072
8	1.563	1.282	0.831	0.774	0.404	0.442	0.099	0.114
16	1.734	1.979	0.907	1.174	0.401	0.683	0.101	0.184
32	1.742	2.957	0.917	1.936	0.386	1.122	0.105	0.310

4.3 Количество повторных передач

Таблица 3: Среднее число повторных передач

W	$p = 0{,}05$		$p = 0{,}10$		$p = 0{,}20$		$p = 0{,}50$	
	GBN	SR	GBN	SR	GBN	SR	GBN	SR
1	110	110	242	242	565	565	2930	2930
2	113	110	257	241	575	565	2583	2930
4	216	110	434	242	1005	565	4078	2921
8	430	110	895	242	1925	564	8051	2907
16	854	110	1735	241	3976	566	15779	2922
32	1788	110	3421	242	8210	564	30202	2883

5 Графики

5.1 Зависимость эффективности от вероятности потери

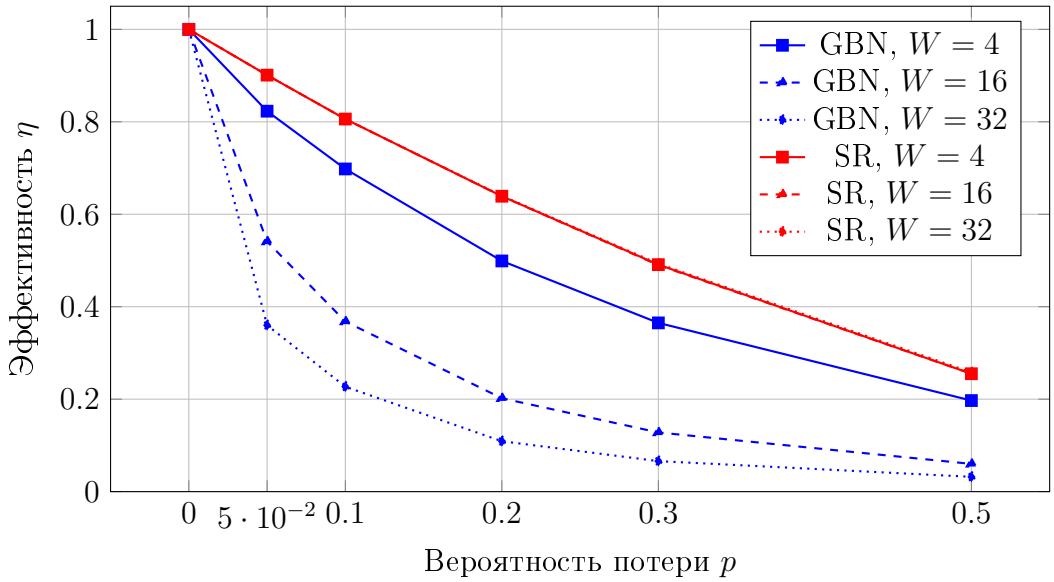

Рис. 1: Эффективность протоколов GBN и SR в зависимости от вероятности потери

5.2 Зависимость пропускной способности от размера окна

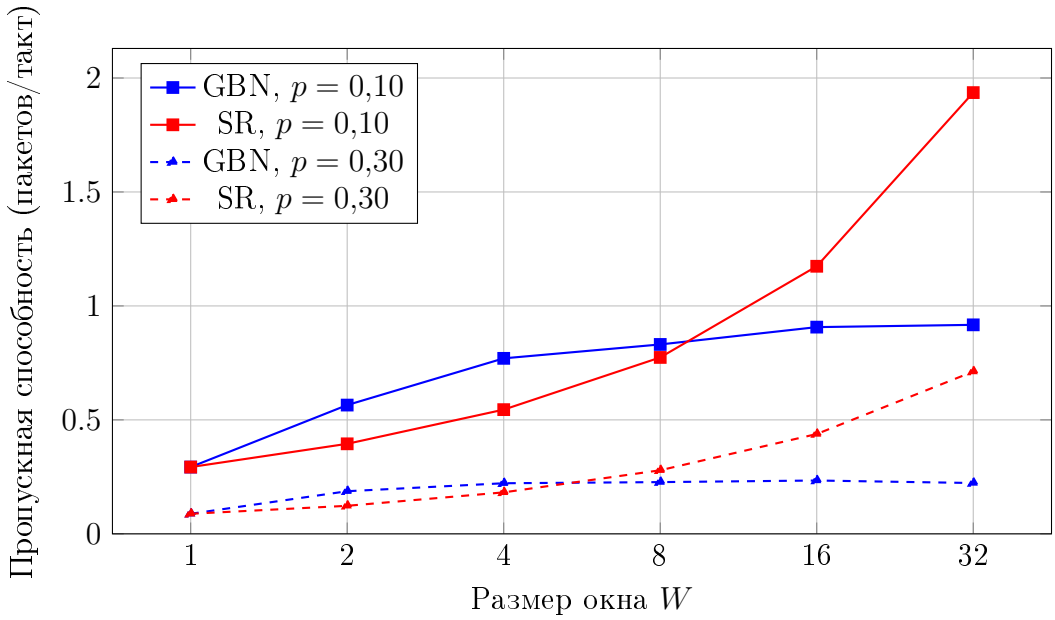

Рис. 2: Пропускная способность в зависимости от размера окна

5.3 Количество повторных передач

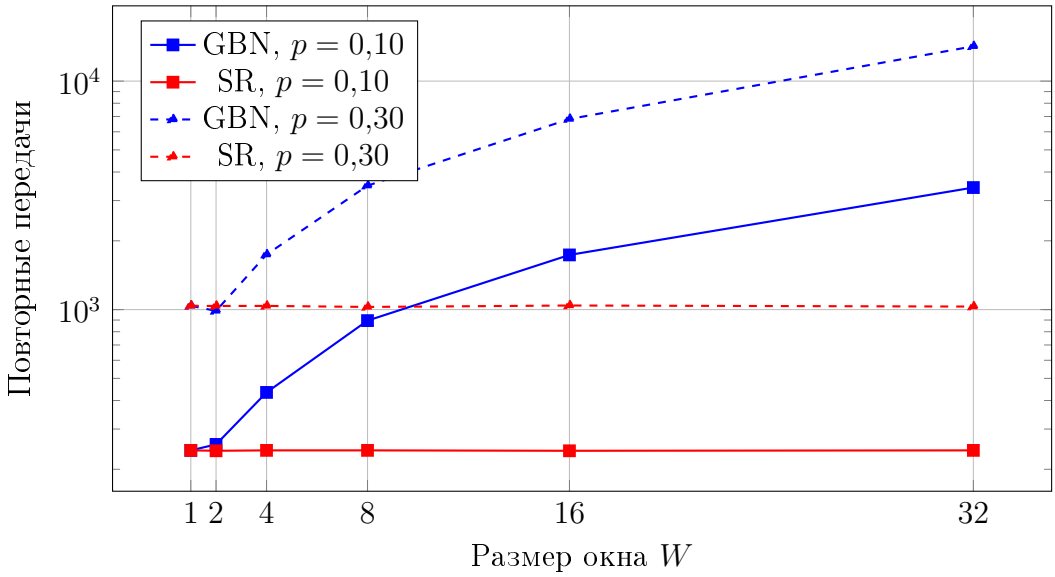

Рис. 3: Количество повторных передач (логарифмическая шкала)

6 Анализ результатов

6.1 Влияние размера окна

GBN: При увеличении размера окна эффективность GBN *снижается* при наличии потерь. Это объясняется тем, что при тайм-ауте GBN повторно передаёт все W пакетов в окне. Чем больше W, тем больше избыточных повторных передач. Например, при $p = 0{,}10$ и $W = 32$ эффективность составляет лишь 0,227, тогда как при $W = 4$ — 0,698.

SR: Эффективность SR *не зависит* от размера окна, так как повторно передаётся только потерянный пакет. При $p = 0{,}10$ эффективность стабильно составляет $\approx 0{,}806$ для любого W.

6.2 Влияние вероятности потери

При увеличении вероятности потери оба протокола теряют эффективность, однако GBN деградирует значительно быстрее, особенно при больших окнах:

- При $p = 0{,}50$ и $W = 32$: GBN отправляет в среднем 31 202 пакетов для передачи 1000 уникальных (эффективность 3,2%), в то время как SR — 3883 пакета (эффективность 25,8%).

6.3 Пропускная способность

Для GBN пропускная способность достигает плато при увеличении W, так как выигрыш от параллельной передачи нивелируется ростом повторных передач. Для SR пропускная способность растёт с увеличением W, однако рост является *сублинейным*: при $p = 0{,}10$ увеличение окна с $W = 1$ до $W = 32$ даёт рост пропускной способности лишь в $\approx 6{,}6$ раз (с 0,293 до 1,936), а не в 32 раза. Это объясняется тем, что при больших окнах вероятность потерять хотя бы один пакет из окна стремится к единице, и отправитель регулярно простаивает в ожидании тайм-аута.

6.4 Случай $W = 1$

При $W = 1$ оба протокола эквивалентны протоколу Stop-and-Wait, что подтверждается одинаковыми результатами в экспериментах.

7 Выводы

1. **Selective Repeat превосходит Go-Back-N** по эффективности использования канала при любых условиях с потерями ($p > 0$).
2. **Преимущество SR растёт** с увеличением размера окна и вероятности потери. При $W = 32$ и $p = 0{,}50$ SR в 8 раз эффективнее GBN.
3. **Эффективность SR не зависит от** W, тогда как эффективность GBN обратно пропорциональна размеру окна при наличии потерь.
4. **Пропускная способность SR растёт сублинейно** с увеличением размера окна: при больших W рост замедляется из-за простоев при тайм-аутах, однако SR остаётся предпочтительным для каналов с ненулевой вероятностью ошибок.
5. **GBN проще в реализации**: не требует буфера на приёмной стороне и может быть предпочтительным при низких вероятностях потерь ($p < 0{,}05$) и малых окнах.
6. При $W = 1$ оба протокола вырождаются в Stop-and-Wait и показывают одинаковую производительность.

Приложение: исходный код

Исходный код симулятора доступен в репозитории:
`https://github.com/AleksandrShvartz/NetworksLabs`