

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ПЕТРА
ВЕЛИКОГО

ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ И МЕХАНИКИ
ВЫСШАЯ ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ФИЗИКИ

**Отчет
по лабораторной работе №2
по дисциплине
“Компьютерные сети”**

**Реализация протокола динамической маршрутизации
Open Shortest Path First**

Выполнил:

Студент: Шварц Александр
Группа: 5040102/40201

Принял:

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

Санкт-Петербург
2026 г.

Содержание

1 Введение	2
2 Теоретические основы	2
2.1 Протокол OSPF	2
2.2 Топологии	2
3 Реализация	3
4 Результаты экспериментов	3
4.1 Начальная сходимость	3
4.2 Масштабируемость	3
4.3 Таблицы маршрутизации	4
4.4 Примеры маршрутов	5
4.5 Стохастические разрывы связей	5
4.6 Разрыв и восстановление одного канала	6
5 Анализ	7
6 Выводы	7
Приложение: исходный код	7

1 Введение

Протокол OSPF (Open Shortest Path First) относится к классу link-state протоколов маршрутизации: каждый маршрутизатор собирает информацию о топологии сети и самостоятельно вычисляет таблицу маршрутизации с помощью алгоритма Дейкстры.

Цель работы — реализовать симулятор протокола OSPF, в котором маршрутизаторы обмениваются служебными сообщениями (LSA), строят базу данных о состоянии каналов и выполняют маршрутизацию по кратчайшему пути. Кроме того, исследуется поведение протокола при отказах каналов: одиночных и массовых стохастических.

Эксперименты проводятся на трёх топологиях: линейной, кольцевой и звёздной.

2 Теоретические основы

2.1 Протокол OSPF

Работа протокола включает три этапа. На первом этапе каждый маршрутизатор формирует LSA (Link State Advertisement) — сообщение, содержащее идентификатор маршрутизатора и список его соседей с указанием стоимости каналов. На втором этапе LSA лавинообразно распространяются по сети: при получении нового LSA маршрутизатор сохраняет его в локальную базу данных (LSDB) и ретранслирует всем соседям. После завершения обмена каждый узел располагает полной копией LSDB. На третьем этапе маршрутизатор применяет алгоритм Дейкстры к LSDB и строит дерево кратчайших путей до всех остальных узлов сети.

2.2 Топологии

В работе рассмотрены три топологии:

- **Линейная** — узлы соединены последовательно: $0-1-2-\dots-(N-1)$. Число каналов $N - 1$, диаметр сети $N - 1$.
- **Кольцо** — аналогична линейной, но с дополнительным каналом между узлами 0 и $N-1$. Число каналов N , диаметр $\lfloor N/2 \rfloor$. Между любой парой узлов существуют два независимых пути.
- **Звезда** — выделенный центральный маршрутизатор (узел 0) соединён со всеми остальными. Число каналов $N - 1$, диаметр 2.

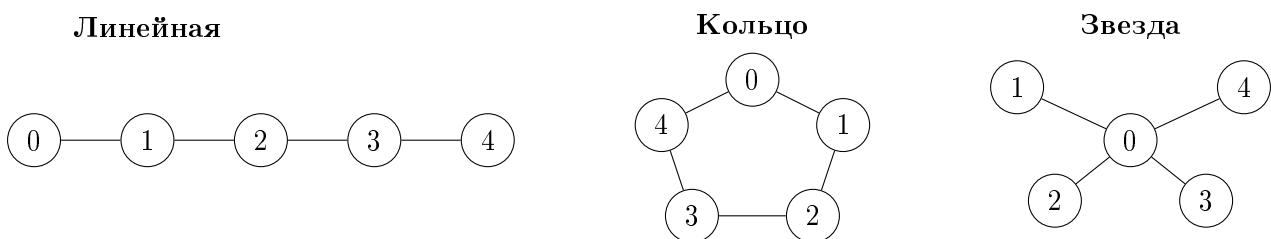


Рис. 1: Топологии сети (пример для $N = 5$)

3 Реализация

Симулятор реализован на языке C++ и включает следующие компоненты:

- **Router** — маршрутизатор с локальной LSDB; генерирует собственные LSA, обрабатывает входящие и вычисляет таблицу маршрутизации алгоритмом Дейкстры;
- **Network** — управляет множеством маршрутизаторов и каналов связи, реализует цикл сходимости и операции разрыва/восстановления каналов;
- **LSA** — структура, содержащая идентификатор источника, порядковый номер и список соседей;
- **Link** — канал между двумя узлами с флагом активности.

Моделирование выполняется пошагово: на каждом такте маршрутизаторы передают соседям все новые LSA. Сходимость фиксируется, когда за полный такт ни одна LSDB не обновилась; после этого каждый узел пересчитывает таблицу маршрутизации. Для верификации результатов после каждого эксперимента выполняется проверка: для всех пар узлов BFS-ом определяется эталонный кратчайший путь и сравнивается с результатом алгоритма Дейкстры.

4 Результаты экспериментов

4.1 Начальная сходимость

В таблице 1 приведены результаты начальной сходимости для $N = 10$ без отказов каналов.

Таблица 1: Начальная сходимость ($N = 10$)

Топология	Тактов до сходимости	Достигимых пар	Средняя длина пути
Линейная	10	90/90	3.67
Кольцо	6	90/90	2.78
Звезда	3	90/90	1.80

Скорость сходимости определяется диаметром сети. В линейной топологии LSA от крайнего узла должно пройти через $N - 1$ промежуточных маршрутизаторов, что даёт N тактов (включая завершающий такт без обновлений). В звезде любое LSA достигает центрального узла за один такт, а от него — всех периферийных за второй, поэтому сходимость составляет 3 такта при любом N .

4.2 Масштабируемость

Зависимость времени сходимости от числа маршрутизаторов представлена в таблице 2 и на рис. 2.

Таблица 2: Время сходимости и средняя длина пути для разных N

N	Линейная		Кольцо		Звезда	
	Такты	Ср. путь	Такты	Ср. путь	Такты	Ср. путь
5	5	2.00	3	1.50	3	1.60
10	10	3.67	6	2.78	3	1.80
20	20	7.00	11	5.26	3	1.90
50	50	17.00	26	12.76	3	1.96

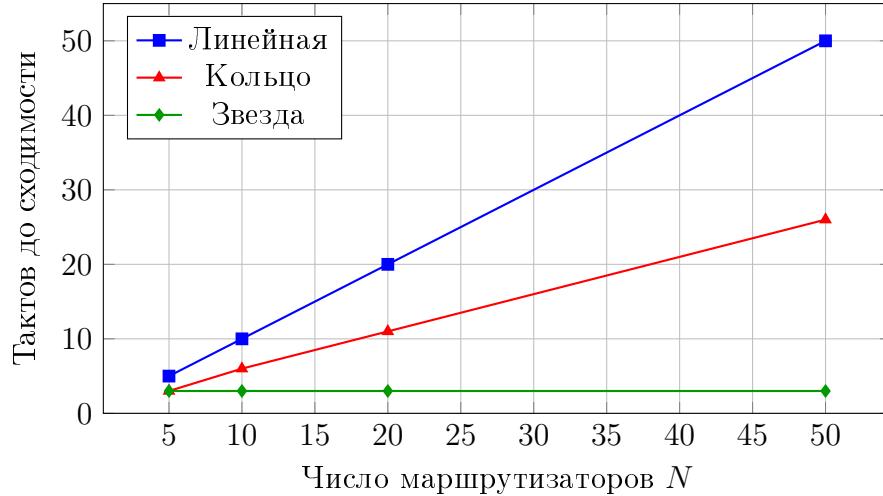


Рис. 2: Время сходимости OSPF в зависимости от N

Для линейной топологии время сходимости растёт как $O(N)$, для кольца — как $O(N/2)$, что соответствует диаметру. Звезда сохраняет постоянное время сходимости (3 такта) при любом числе узлов.

4.3 Таблицы маршрутизации

В таблице 3 приведены таблицы маршрутизации узла 0 для каждой топологии при $N = 10$.

Таблица 3: Таблица маршрутизации узла 0

Назначение	Линейная		Кольцо		Звезда	
	Сл. хоп	Цена	Сл. хоп	Цена	Сл. хоп	Цена
1	1	1	1	1	1	1
2	1	2	1	2	2	1
3	1	3	1	3	3	1
4	1	4	1	4	4	1
5	1	5	1	5	5	1
6	1	6	9	4	6	1
7	1	7	9	3	7	1
8	1	8	9	2	8	1
9	1	9	9	1	9	1

В линейной топологии единственным следующим хопом для узла 0 является узел 1. В кольце маршруты разделяются: к узлам 1–5 трафик направляется через узел 1, к узлам

6–9 — через узел 9, что соответствует выбору кратчайшего направления. В звезде узел 0 является центральным, поэтому все назначения доступны за один хоп.

4.4 Примеры маршрутов

Таблица 4: Маршруты от узла 0 к узлам 5 и 9

Топология	$0 \rightarrow 5$	$0 \rightarrow 9$
Линейная	$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$	$0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 9$
Кольцо	$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$	$0 \rightarrow 9$
Звезда	$0 \rightarrow 5$	$0 \rightarrow 9$

Маршрут $0 \rightarrow 9$ в кольце проходит в один хоп, что подтверждает корректность выбора кратчайшего направления алгоритмом Дейкстры.

4.5 Стохастические разрывы связей

В данном эксперименте каждый канал сети независимо выходит из строя с вероятностью p . После отказов маршрутизаторы выполняют повторную сходимость, после чего измеряется доля достижимых пар узлов. Результаты усреднены по 20 запускам.

Таблица 5: Стохастические разрывы ($N = 10$)

p	Линейная (9 кан.)			Кольцо (10 кан.)			Звезда (9 кан.)		
	Разр.	Связн.	Такты	Разр.	Связн.	Такты	Разр.	Связн.	Такты
0.00	0.0	1.00	10.0	0.0	1.00	6.0	0.0	1.00	3.0
0.10	0.9	0.73	8.3	0.9	0.92	8.2	0.9	0.84	3.0
0.20	1.6	0.59	7.2	1.7	0.72	6.9	1.6	0.71	3.0
0.30	2.5	0.43	5.8	2.8	0.52	5.8	2.5	0.58	3.0
0.50	4.3	0.23	3.9	4.8	0.27	4.1	4.3	0.35	3.0

Разр. — среднее число разорванных каналов; **Связн.** — доля достижимых пар; **Такты** — время пересходимости.

Во всех 300 запусках (3 топологии \times 5 вероятностей \times 20 повторов) таблицы маршрутизации после повторной сходимости прошли верификацию — расхождений с эталонными BFS-маршрутами не обнаружено.

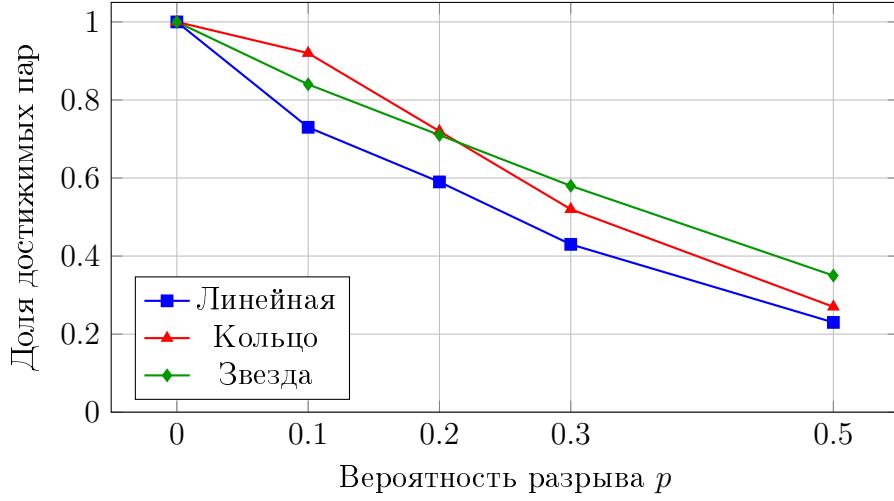


Рис. 3: Связность сети при стохастических разрывах

При малых вероятностях отказа наибольшую устойчивость демонстрирует кольцо (92% при $p = 0,1$), поскольку между любой парой узлов существуют два независимых пути. Линейная топология наиболее уязвима: единственный разрыв канала разделяет сеть на два сегмента, что даёт 73% связности уже при $p = 0,1$.

При $p = 0,5$ звезда превосходит кольцо по связности (35% против 27%). Это объясняется тем, что в звезде связность каждого периферийного узла зависит только от одного канала до центра, тогда как в кольце множественные разрывы приводят к фрагментации сети на изолированные сегменты.

4.6 Разрыв и восстановление одного канала

В данном эксперименте моделируется разрыв и последующее восстановление канала 0–1.

Таблица 6: Разрыв и восстановление канала 0–1

Топология	Событие	Тактов	Достижимых пар	Корректность
Линейная	Начальное состояние	10	90	Да
	Разрыв	9	72	Да
	Восстановление	10	90	Да
Кольцо	Начальное состояние	6	90	Да
	Разрыв	10	90	Да
	Восстановление	6	90	Да
Звезда	Начальное состояние	3	90	Да
	Разрыв	3	72	Да
	Восстановление	3	90	Да

В кольце после разрыва канала 0–1 все 90 пар остаются достижимыми — трафик перенаправляется по альтернативному пути. При этом топология фактически вырождается в линейную, и время сходимости возрастает с 6 до 10 тактов. В линейной топологии и в звезде разрыв канала 0–1 изолирует узел 1 от части сети, в результате чего 18 пар узлов становятся недостижимыми.

5 Анализ

Результаты экспериментов позволяют сопоставить характеристики трёх топологий.

Звезда обеспечивает наименьшее время сходимости (3 такта при любом N) и кратчайшие маршруты, однако полностью зависит от центрального узла: его отказ приводит к полной потере связности.

Кольцо обладает избыточностью: одиночный разрыв канала не нарушает связность, поскольку между любой парой узлов существует альтернативный путь. Время сходимости составляет $\approx N/2$ тактов. Вместе с тем при множественных отказах устойчивость кольца снижается.

Линейная топология наиболее уязвима к отказам: любой единственный разрыв канала разделяет сеть на два несвязных сегмента.

По устойчивости к массовым сбоям при $p = 0,1$ кольцо сохраняет 92% связности, звезда — 84%, линейная — 73%. При $p = 0,3$ все три топологии опускаются ниже 60%, что свидетельствует о необходимости избыточных каналов связи в реальных сетях.

6 Выводы

1. Реализован симулятор протокола OSPF на C++ с механизмом рассылки LSA, алгоритмом Дейкстры и пошаговой маршрутизацией. Корректность таблиц маршрутизации подтверждена сравнением с эталонными BFS-маршрутами во всех экспериментах.
2. Время сходимости определяется диаметром сети: N тактов для линейной топологии, $\approx N/2$ для кольца, 3 такта для звезды независимо от числа узлов.
3. Кольцо — единственная из рассмотренных топологий, сохраняющая полную связность при одиночном разрыве канала.
4. Звезда обеспечивает минимальное время сходимости и кратчайшие маршруты, однако представляет собой единую точку отказа.
5. При стохастических разрывах с $p \geq 0,3$ все три топологии теряют более 40% связности, что указывает на необходимость использования избыточных топологий (ячеистых, полносвязных) в реальных сетях.

Приложение: исходный код

Исходный код симулятора доступен в репозитории:

<https://github.com/AleksandrShvartz/NetworksLabs>