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

В данной работе рассматривается задача динамической маршрутизации в компьютерных сетях. За основу взят протокол OSPF (Open Shortest Path First), который относится к классу link-state протоколов: каждый маршрутизатор собирает информацию о всей топологии сети и на её основе самостоятельно строит таблицу маршрутизации, используя алгоритм Дейкстры.

Задача состоит в том, чтобы написать симулятор, в котором несколько маршрутизаторов обмениваются служебными сообщениями, узнают структуру сети и умеют пересылать данные по кратчайшему пути. Помимо этого, нужно посмотреть, как протокол справляется с отказами каналов — как при одиночном разрыве, так и при случайных массовых сбоях.

Рассматриваются три топологии: линейная цепочка, кольцо и звезда.

2 Теоретические основы

2.1 Как работает OSPF

Работа протокола разбивается на три фазы. Сначала каждый маршрутизатор составляет так называемое LSA (Link State Advertisement) — по сути, список своих соседей с указанием стоимости канала до каждого из них. Затем эти LSA лавинообразно рассылаются по сети: получив чужое LSA, маршрутизатор запоминает его и пересылает дальше всем своим соседям. Когда новых LSA больше не приходит, у каждого узла есть полная база данных о состоянии каналов (LSDB). На третьем шаге каждый маршрутизатор запускает алгоритм Дейкстры на этой базе и получает дерево кратчайших путей до всех остальных узлов.

2.2 Топологии

В работе рассмотрены три варианта соединения маршрутизаторов:

- **Линейная** — узлы стоят цепочкой: $0{-}1{-}2{-}\ldots{-}(N{-}1)$. Каналов $N-1$, диаметр сети $N-1$. Самая простая и самая уязвимая к разрывам.
- **Кольцо** — то же, но последний узел соединён с первым: $0{-}1{-}\ldots{-}(N{-}1){-}0$. Каналов N, диаметр $\lfloor N/2 \rfloor$. Появляется второй путь между любыми узлами.
- **Звезда** — один центральный маршрутизатор (0), к которому подключены все остальные. Каналов $N-1$, диаметр 2. Быстрая сходимость, но если центр откажет — всё рассыплется.

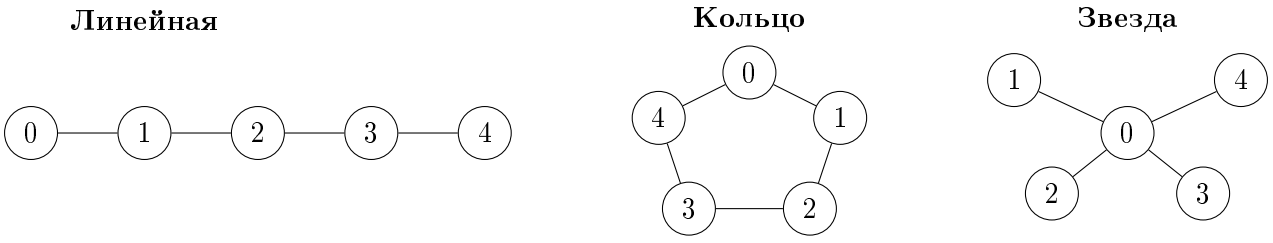

Рис. 1: Топологии сети (пример для $N = 5$)

3 Реализация

Программа написана на C++. Основные классы:

- `Router` — хранит свою LSDB, умеет генерировать LSA, принимать чужие и запускать Дейкстру для построения таблицы маршрутизации;
- `Network` — собирает маршрутизаторы и каналы воедино, запускает процесс сходимости (итеративный обмен LSA до стабилизации) и умеет ломать/чинить каналы;
- `LSA` — структура с id источника, порядковым номером и списком соседей;
- `Link` — канал между двумя узлами, у которого есть флаг `active`.

Сходимость устроена просто: на каждом такте маршрутизаторы пересылают соседям все новые LSA. Как только за целый такт ни одна LSDB не обновилась — считаем, что сеть сошлась, и каждый узел пересчитывает свою таблицу. Чтобы убедиться в правильности, после каждого эксперимента таблицы маршрутизации проверяются: для каждой пары узлов BFS-ом находится эталонный кратчайший путь и сравнивается с тем, что выдал Дейкстра.

4 Результаты экспериментов

4.1 Начальная сходимость

Первым делом посмотрим, как быстро протокол сходится на каждой топологии при 10 маршрутизаторах и без каких-либо сбоев.

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

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

Результат ожидаемый: скорость сходимости напрямую зависит от диаметра сети. В линейной топологии самый дальний LSA должен пройти через 9 промежуточных узлов — отсюда 10 тактов (9 на рассылку + 1 проверочный, когда уже ничего нового не приходит). В звезде любой LSA за один шаг попадает в центр, а оттуда за второй шаг — куда угодно, поэтому 3 тактов хватает при любом N.

4.2 Масштабируемость

Теперь увеличим число маршрутизаторов и посмотрим, как ведёт себя время сходимости.

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

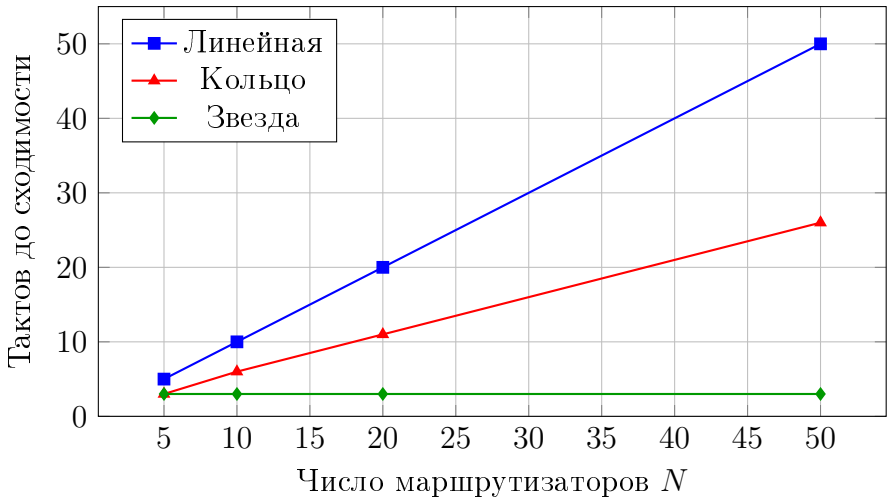

Рис. 2: Время сходимости OSPF в зависимости от N

Хорошо видно: для линейной топологии время сходимости растёт строго как N, для кольца — примерно как $N/2$ (что совпадает с диаметром), а звезда остаётся на 3 тактах даже при 50 узлах.

4.3 Таблицы маршрутизации

Для наглядности приведём таблицу маршрутизации узла 0 в каждой топологии ($N = 10$).

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 у узла 0 выбора нет — всё идёт через хоп 1. В кольце интереснее: до узлов 1–5 короче идти «по часовой» (через 1), а до узлов 6–9 — «против часовой» (через

9). В звезде узел 0 и есть центр, поэтому каждый пункт назначения доступен напрямую за 1 хоп.

4.4 Примеры маршрутов

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

Топология	$0 \to 5$	$0 \to 9$
Линейная	$0 \to 1 \to 2 \to 3 \to 4 \to 5$	$0 \to 1 \to 2 \to \ldots \to 9$
Кольцо	$0 \to 1 \to 2 \to 3 \to 4 \to 5$	$0 \to 9$
Звезда	$0 \to 5$	$0 \to 9$

Маршрут $0 \to 9$ в кольце проходит в один хоп — это показывает, что Дейкстра действительно выбирает более короткое направление.

4.5 Стохастические разрывы связей

Самый интересный эксперимент: берём рабочую сеть и случайным образом рвём каналы — каждый независимо с вероятностью p. После этого маршрутизаторы пересходятся, и мы смотрим, какая доля пар узлов осталась достижимой. Результаты усреднены по 20 запускам.

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

	Линейная (9 кан.)			Кольцо (10 кан.)			Звезда (9 кан.)		
p	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.

Важный момент: во всех 300 запусках (3 топологии $\times$ 5 вероятностей $\times$ 20 повторов) таблицы после пересходимости оказались корректными — протокол ни разу не «заблудился».

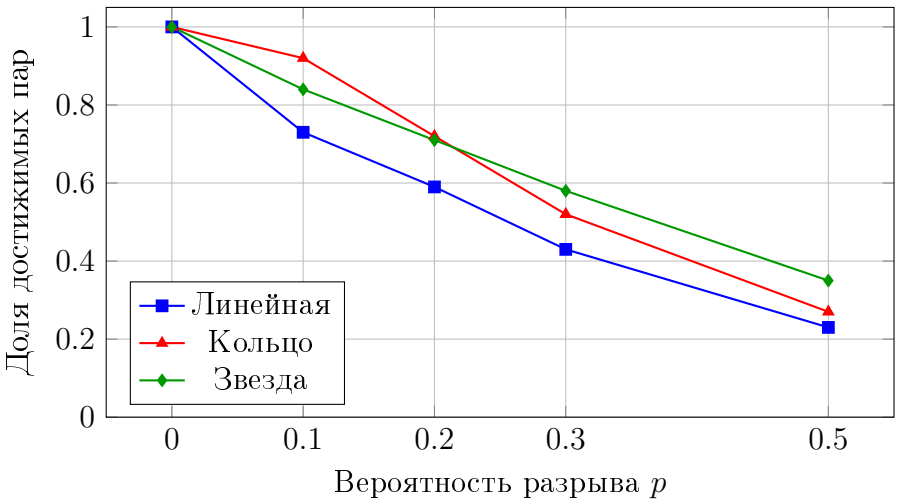

Рис. 3: Связность сети при стохастических разрывах

При небольших p лучше всех держится кольцо (92% при $p = 0{,}1$) — логично, ведь там два пути между любыми узлами, и для потери связности нужно оборвать оба. Линейная топология теряет связность уже при одном разрыве, поэтому проседает сильнее всего (73%).

Любопытно, что при $p = 0{,}5$ звезда обгоняет кольцо (35% против 27%). Дело в том, что в звезде каждый периферийный узел зависит только от своего единственного канала до центра, а в кольце при половине оборванных каналов сеть рассыпается на множество мелких фрагментов.

4.6 Разрыв и восстановление одного канала

Отдельно проверим поведение при поломке и починке конкретного 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

Здесь хорошо видна разница между топологиями. В кольце после разрыва канала 0–1 все 90 пар по-прежнему достижимы — трафик просто пошёл в обход. Правда, сеть фактически стала линейной, и время пересходимости подскочило с 6 до 10 тактов. В линейной топологии и в звезде разрыв канала 0–1 отрезает узел 1, и 18 пар (с участием узла 1) становятся недостижимыми.

5 Анализ

У каждой топологии свои сильные и слабые стороны.

Звезда — чемпион по скорости: сходится за 3 такта при любом N, маршруты самые короткие. Но всё завязано на центральный узел. Если он упадёт, сеть перестаёт существовать. Для небольших сетей с надёжным центром это хороший выбор.

Кольцо — золотая середина. Единичный разрыв канала не страшен: всегда есть обходной путь. Время сходимости умеренное ($\approx N/2$). Из минусов — при двух и более разрывах всё уже не так радужно.

Линейная топология — самая хрупкая. Любой единственный разрыв рубит сеть на две части. На практике чистая линейная топология встречается разве что в цепочках повторителей.

Что касается устойчивости к массовым сбоям: при $p = 0{,}1$ кольцо сохраняет 92% связности, звезда 84%, линейная — только 73%. При $p = 0{,}3$ все топологии уже ниже 60%, что для реальной сети означало бы серьёзные проблемы.

6 Выводы

1. Реализован симулятор протокола OSPF на C++ с рассылкой LSA, алгоритмом Дейкстры и маршрутизацией по хопам. Корректность проверена на всех экспериментах.

2. Скорость сходимости определяется диаметром сети: N тактов для линейной, $\approx N/2$ для кольца, 3 такта для звезды вне зависимости от размера.

3. Кольцо — единственная из рассмотренных топологий, которая переживает одиночный разрыв без потери связности.

4. Звезда быстрее всех сходится и даёт кратчайшие маршруты, но полностью зависит от центрального узла.

5. При случайных разрывах с $p \geq 0{,}3$ все три топологии теряют больше 40% связности — в реальных сетях это аргумент в пользу избыточных каналов и более сложных топологий (например, полносвязных или ячеистых).

Приложение: исходный код

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