

Санкт-Петербургский политехнический университет
Петра Великого

Физико-механический институт

Кафедра «Прикладная математика»

**Отчёт по лабораторной работе
по дисциплине «Компьютерные сети»
Реализация протокола маршрутизации Open Shortest Path
First**

Выполнил студент:
Габдрахманов Булат Маратович
группа: 5040102/20201

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

Санкт-Петербург
2023 г.

Содержание

1	Постановка задачи	2
2	Теория	2
3	Реализация	2
4	Результаты	2
5	Заключение	11

1 Постановка задачи

Требуется выполнить реализацию протокола маршрутизации OSPF (Open Shortest Path First) и осуществить проверку его функционирования в различных конфигурациях сетевых топологий, включая линейную, кольцевую и звездообразную.

2 Теория

OSPF (Open Shortest Path First) — это протокол динамической маршрутизации, работающий на основе мониторинга состояния связей и применяющий алгоритм Дейкстры для определения наиболее коротких маршрутов.

Основные принципы работы протокола:

- Сначала маршрутизаторы, после их активации, ищут соседние устройства, непосредственно подключенные к ним, и налаживают с ними взаимодействие.
- Затем маршрутизаторы обмениваются сведениями о сетях, к которым они имеют доступ, создавая общую карту (топологию) сети. Эта карта является одинаковой для всех устройств в сети.
- На последнем этапе активируется алгоритм SPF (Shortest Path First), целью которого является вычисление наилучшего маршрута до каждой из сетей. Этот процесс можно сравнить с построением дерева, где маршрутизатор является корнем, а маршруты к различным сетям - ветвями.

3 Реализация

Весь код написан на языке Python (версии 3.9). Для каждого протокола получатель и отправитель работают параллельно в отдельных потоках. [Ссылка на GitHub с исходным кодом.](#)

4 Результаты

Изначально проанализируем работу протокола в сетевой структуре с линейной топологией. Обратим внимание на размещение узлов внутри такой сети.

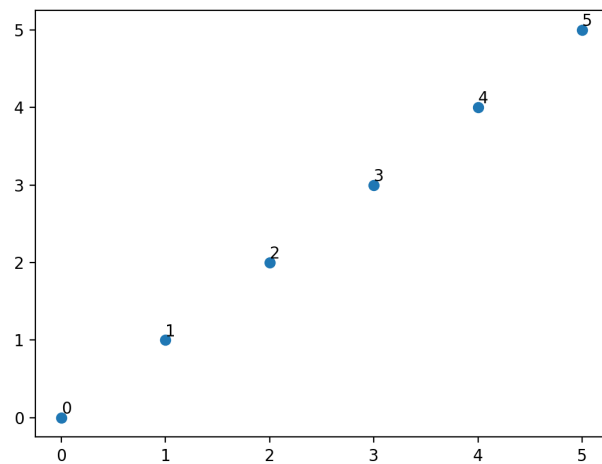


Рис. 1: Расположение узлов сети с линейной топологией

Построим граф сети и установим радиус соединения равным $r = 1.5$.

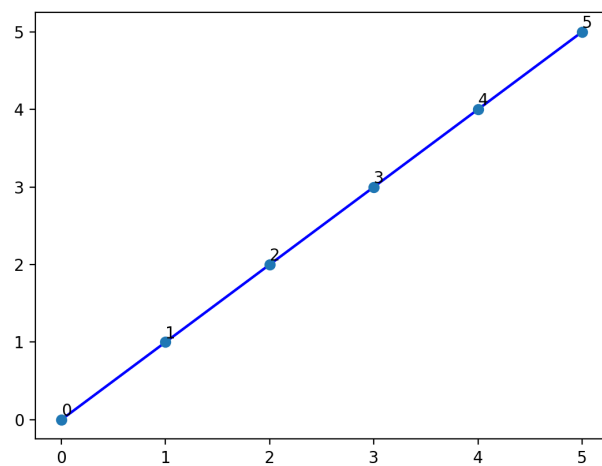


Рис. 2: Граф сети с линейной топологией

Найдём кратчайшие пути между всеми парами узлов сети. Приведём некоторые примеры (полный список результатов можно найти в файле `2_lab/results/line_full.txt`).

- Начальный узел 0
 - путь 0 -> 1: [0, 1]
 - путь 0 -> 2: [0, 1, 2]
 - путь 0 -> 3: [0, 1, 2, 3]
 - путь 0 -> 4: [0, 1, 2, 3, 4]
 - путь 0 -> 5: [0, 1, 2, 3, 4, 5]
- Начальный узел 4
 - путь 4 -> 0: [4, 3, 2, 1, 0]
 - путь 4 -> 1: [4, 3, 2, 1]
 - путь 4 -> 2: [4, 3, 2]
 - путь 4 -> 3: [4, 3]
 - путь 4 -> 5: [4, 5]

Теперь исключим узел 3 из сети, переместив его на значительное расстояние, и затем перерисуем граф сети с учётом этого изменения.

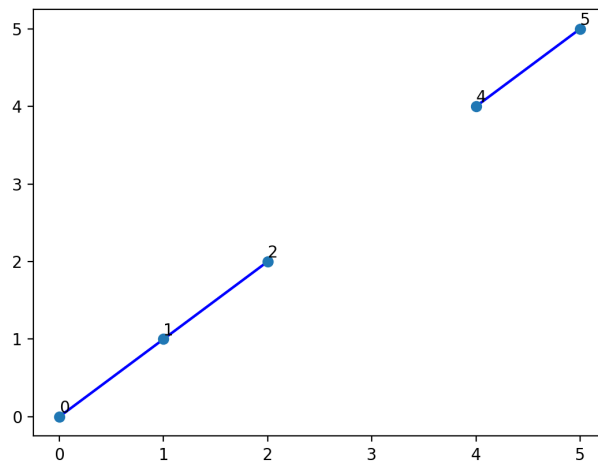


Рис. 3: Граф сети с линейной топологией без 3 узла

Укажем кратчайшие пути для тех же пар узлов, что и ранее (полный список результатов можно найти в файле `2_lab/results/line_remove.txt`).

- Начальный узел 0
 - путь 0 \rightarrow 1: $[0, 1]$
 - путь 0 \rightarrow 2: $[0, 1, 2]$
 - путь 0 \rightarrow 3: $[]$
 - путь 0 \rightarrow 4: $[]$
 - путь 0 \rightarrow 5: $[]$
- Начальный узел 4
 - путь 4 \rightarrow 0: $[]$
 - путь 4 \rightarrow 1: $[]$
 - путь 4 \rightarrow 2: $[]$
 - путь 4 \rightarrow 3: $[]$
 - путь 4 \rightarrow 5: $[4, 5]$

Проведём аналогичную процедуру для сети с кольцевидной топологией.

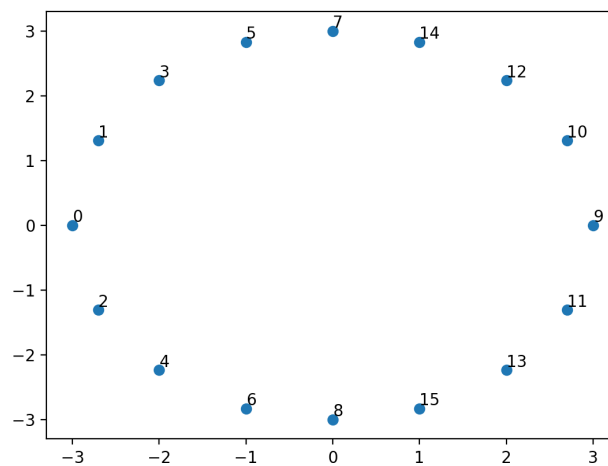


Рис. 4: Расположение узлов сети с кольцевидной топологией

Граф, построенный с радиусом соединения $r = 1.7$, сети имеет вид.

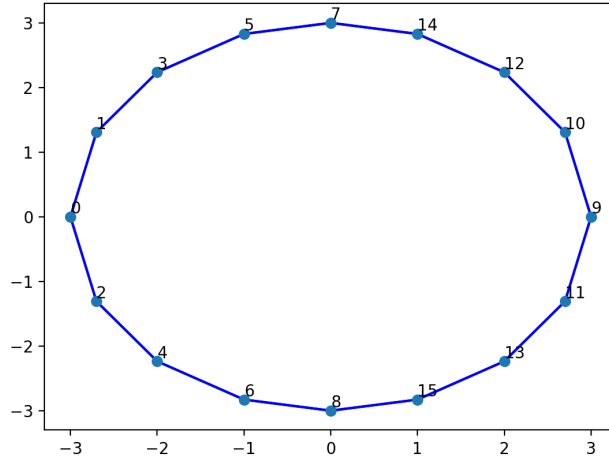


Рис. 5: Граф сети с кольцевидной топологией

Примеры кратчайших путей (подробнее [2_lab/results/ring_full.txt](#))

- Начальный узел 5
 - путь 5 -> 0: [5, 3, 1, 0]
 - путь 5 -> 1: [5, 3, 1]
 - путь 5 -> 2: [5, 3, 1, 0, 2]
 - путь 5 -> 3: [5, 3]
 - путь 5 -> 4: [5, 3, 1, 0, 2, 4]
 - путь 5 -> 6: [5, 3, 1, 0, 2, 4, 6]
 - путь 5 -> 7: [5, 7]
 - путь 5 -> 8: [5, 3, 1, 0, 2, 4, 6, 8]
 - путь 5 -> 9: [5, 7, 14, 12, 10, 9]
 - путь 5 -> 10: [5, 7, 14, 12, 10]
 - путь 5 -> 11: [5, 7, 14, 12, 10, 9, 11]
 - путь 5 -> 12: [5, 7, 14, 12]
 - путь 5 -> 13: [5, 7, 14, 12, 10, 9, 11, 13]
 - путь 5 -> 14: [5, 7, 14]
 - путь 5 -> 15: [5, 7, 14, 12, 10, 9, 11, 13, 15]

- Начальный узел 12
 - путь 12 \rightarrow 0: [12, 14, 7, 5, 3, 1, 0]
 - путь 12 \rightarrow 1: [12, 14, 7, 5, 3, 1]
 - путь 12 \rightarrow 2: [12, 14, 7, 5, 3, 1, 0, 2]
 - путь 12 \rightarrow 3: [12, 14, 7, 5, 3]
 - путь 12 \rightarrow 4: [12, 14, 7, 5, 3, 1, 0, 2, 4]
 - путь 12 \rightarrow 5: [12, 14, 7, 5]
 - путь 12 \rightarrow 6: [12, 10, 9, 11, 13, 15, 8, 6]
 - путь 12 \rightarrow 7: [12, 14, 7]
 - путь 12 \rightarrow 8: [12, 10, 9, 11, 13, 15, 8]
 - путь 12 \rightarrow 9: [12, 10, 9]
 - путь 12 \rightarrow 10: [12, 10]
 - путь 12 \rightarrow 11: [12, 10, 9, 11]
 - путь 12 \rightarrow 13: [12, 10, 9, 11, 13]
 - путь 12 \rightarrow 14: [12, 14]
 - путь 12 \rightarrow 15: [12, 10, 9, 11, 13, 15]

После удаления узла 11 граф сети имеет вид.

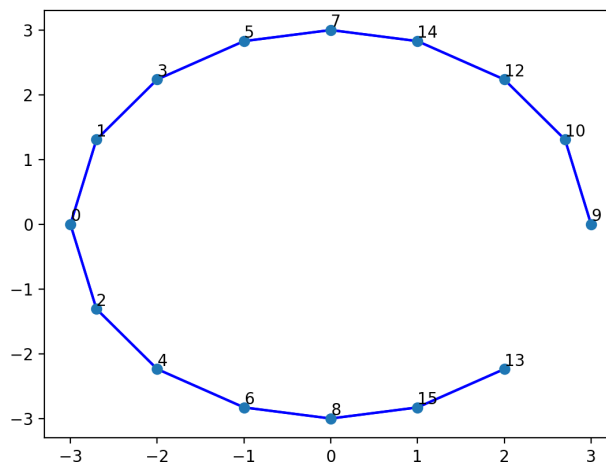


Рис. 6: Граф сети с кольцевидной топологией без 11 узла

Примеры путей для тех же пар узлов (подробнее *2_lab/results/ring_remove.txt*)

- Начальный узел 5

- путь 5 -> 0: [5, 3, 1, 0]
- путь 5 -> 1: [5, 3, 1]
- путь 5 -> 2: [5, 3, 1, 0, 2]
- путь 5 -> 3: [5, 3]
- путь 5 -> 4: [5, 3, 1, 0, 2, 4]
- путь 5 -> 6: [5, 3, 1, 0, 2, 4, 6]
- путь 5 -> 7: [5, 7]
- путь 5 -> 8: [5, 3, 1, 0, 2, 4, 6, 8]
- путь 5 -> 9: [5, 7, 14, 12, 10, 9]
- путь 5 -> 10: [5, 7, 14, 12, 10]
- путь 5 -> 11: []
- путь 5 -> 12: [5, 7, 14, 12]
- путь 5 -> 13: [5, 3, 1, 0, 2, 4, 6, 8, 15, 13]
- путь 5 -> 14: [5, 7, 14]
- путь 5 -> 15: [5, 3, 1, 0, 2, 4, 6, 8, 15]

- Начальный узел 12

- путь 12 -> 0: [12, 14, 7, 5, 3, 1, 0]
- путь 12 -> 1: [12, 14, 7, 5, 3, 1]
- путь 12 -> 2: [12, 14, 7, 5, 3, 1, 0, 2]
- путь 12 -> 3: [12, 14, 7, 5, 3]
- путь 12 -> 4: [12, 14, 7, 5, 3, 1, 0, 2, 4]
- путь 12 -> 5: [12, 14, 7, 5]
- путь 12 -> 6: [12, 14, 7, 5, 3, 1, 0, 2, 4, 6]
- путь 12 -> 7: [12, 14, 7]
- путь 12 -> 8: [12, 14, 7, 5, 3, 1, 0, 2, 4, 6, 8]
- путь 12 -> 9: [12, 10, 9]
- путь 12 -> 10: [12, 10]
- путь 12 -> 11: []

- путь 12 \rightarrow 13: [12, 14, 7, 5, 3, 1, 0, 2, 4, 6, 8, 15, 13]
- путь 12 \rightarrow 14: [12, 14]
- путь 12 \rightarrow 15: [12, 14, 7, 5, 3, 1, 0, 2, 4, 6, 8, 15]

Узлы сети со звёздной топологией и центральным узлом 0 имеют следующее расположение.

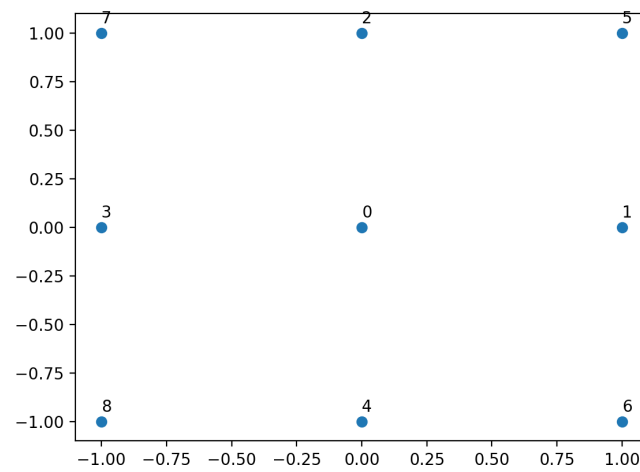


Рис. 7: Расположение узлов сети с звёздной топологией

Граф для данной сети имеет вид.

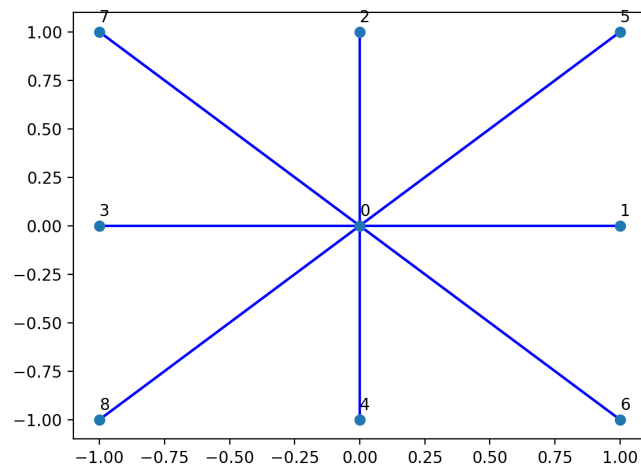


Рис. 8: Граф сети с звёздной топологией

Некоторые примеры кратчайший путей (подробнее [2_lab/results/star_full.txt](#)).

- Начальный узел 0
 - путь 0 -> 1: [0, 1]
 - путь 0 -> 2: [0, 2]
 - путь 0 -> 3: [0, 3]
 - путь 0 -> 4: [0, 4]
 - путь 0 -> 5: [0, 5]
 - путь 0 -> 6: [0, 6]
 - путь 0 -> 7: [0, 7]
 - путь 0 -> 8: [0, 8]
- Начальный узел 7
 - путь 7 -> 0: [7, 0]
 - путь 7 -> 1: [7, 0, 1]
 - путь 7 -> 2: [7, 0, 2]
 - путь 7 -> 3: [7, 0, 3]
 - путь 7 -> 4: [7, 0, 4]

- путь 7 \rightarrow 5: [7, 0, 5]
- путь 7 \rightarrow 6: [7, 0, 6]
- путь 7 \rightarrow 8: [7, 0, 8]

После удаления центрального узла 0 граф сети имеет вид.

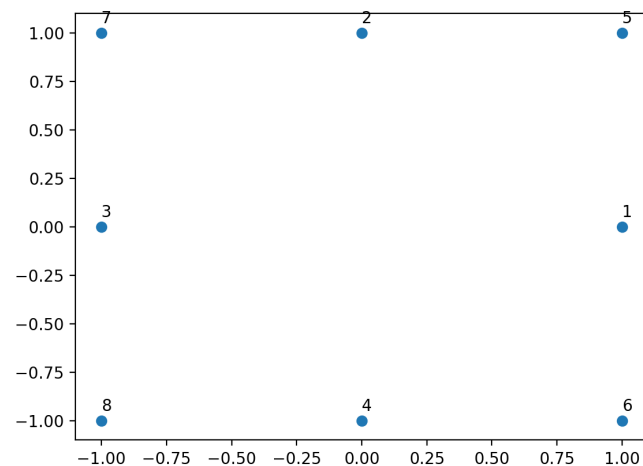


Рис. 9: Граф сети с звёздной топологией без центрального узла 0

Путей для тех же пар узлов (подробнее `2_lab/results/star_remove.txt`) не будет существовать.

5 Заключение

Из анализа результатов становится ясно, что сеть с линейной топологией особенно подвержена влиянию потерь узлов: исчезновение одного узла приводит к появлению узлов, до которых невозможно достучаться. Сеть с кольцевой топологией более устойчива к потере узлов; при пропаже одного из них она преобразуется в сеть линейного типа. Сеть с звёздной топологией наиболее устойчива к потерям узлов, за исключением случаев, когда теряется центральный узел. При этом каждая пара остальных узлов становится недоступной друг для друга.