

Реализация протокола Open Shortest Path First

Данил Пестряков

23 ноября 2022 г.

Содержание

1	Постановка задачи	2
2	Реализация	3
2.1	Описание работы протокола	3
2.2	Выделенный маршрутизатор (DR)	3
3	Результаты	4
3.1	Линейная топология сети	4
3.1.1	Исходное состояние	4
3.1.2	После отключения маршрутизатора с номером 2	4
3.2	Топология сети кольцо	4
3.2.1	Исходное состояние	4
3.2.2	После отключения маршрутизатора с номером 4	6
3.3	Топология сети звезда	6
3.3.1	Исходное состояние	6
3.3.2	После отключения маршрутизатора с номером 4	7
4	Выводы	9
5	Ссылка на github	10
6	Приложения	11

1 Постановка задачи

- Реализовать протокол **OSPF**
- Топология сети: линейная, кольцо, звезда
- Рассмотреть перестройку таблиц достижимости при стохастических разрывах связи

OSPF (англ. Open Shortest Path First) - протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала и использующий для нахождения кратчайшего пути алгоритм Дейкстры.

Протокол OSPF распространяет информацию о доступных маршрутах между маршрутизаторами одной автономной системы.

2 Реализация

2.1 Описание работы протокола

Принцип работы заключается в следующем:

1. После включения маршрутизаторов протокол ищет непосредственно подключённых соседей и устанавливает с ними «дружеские» отношения.
2. Затем они обмениваются друг с другом информацией о подключённых и доступных им сетях. То есть они строят карту сети (топологию сети). Данная карта одинакова на всех маршрутизаторах
3. На основе полученной информации запускается алгоритм SPF (Shortest Path First, «выбор наилучшего пути»), который рассчитывает оптимальный маршрут к каждой сети. Данный процесс похож на построение дерева, корнем которого является сам маршрутизатор, а ветвями — пути к доступным сетям. Данный процесс, то есть конвергенция, происходит очень быстро.

Модель реализована на языке **Python**. Все роутеры работают в отдельных потоках, создаваемых с использованием модуля *threading*.

2.2 Выделенный маршрутизатор (DR)

Выделенный маршрутизатор (designated router, **DR**) — управляет процессом рассылки LSA (link-state advertisement, объявление о состоянии канала) в сети. Каждый маршрутизатор сети устанавливает отношения смежности с DR. Информация об изменениях в сети отправляется маршрутизатором, обнаружившим это изменение, на выделенный маршрутизатор, а тот, в свою очередь, отвечает за то, чтобы эта информация была отправлена остальным маршрутизаторам сегмента множественного доступа.

3 Результаты

3.1 Линейная топология сети

3.1.1 Исходное состояние

Взято 5 маршрутизаторов. Модель сети: 0 - 1 - 2 - 3 - 4

Таблица 1: Таблицы достижимости линейной топологии сети из 5 маршрутизаторов из исходного состояния

Маршрутизатор	Оптимальные маршруты
0	0 0 → 1 0 → 1 → 2 0 → 1 → 2 → 3 0 → 1 → 2 → 3 → 4
1	1 → 0 1 1 → 2 1 → 2 → 3 1 → 2 → 3 → 4
2	2 → 1 → 0 2 → 1 2 2 → 3 2 → 3 → 4
3	3 → 2 → 1 → 0 3 → 2 → 1 3 → 2 3 3 → 4
4	4 → 3 → 2 → 1 → 0 4 → 3 → 2 → 1 4 → 3 → 2 4 → 3 4

3.1.2 После отключения маршрутизатора с номером 2

Модель сети: 0 - 1 3 - 4

3.2 Топология сети кольцо

3.2.1 Исходное состояние

Взято 5 маршрутизаторов. Модель сети: 0 - 1 - 2 - 3 - 4 - 0

Таблица 2: Таблицы достижимости линейной топологии сети из 5 маршрутизаторов после отключения маршрутизатора 2

Маршрутизатор	Оптимальные маршруты
0	0 0 \rightarrow 1
1	1 \rightarrow 0 1
3	3 3 \rightarrow 4
4	4 \rightarrow 3 4

Таблица 3: Таблицы достижимости кольца из 5 маршрутизаторов в исходном состоянии

Маршрутизатор	Оптимальные маршруты
0	0 0 \rightarrow 1 0 \rightarrow 1 \rightarrow 2 0 \rightarrow 4 \rightarrow 3 0 \rightarrow 4
1	1 \rightarrow 0 1 1 \rightarrow 2 1 \rightarrow 2 \rightarrow 3 1 \rightarrow 0 \rightarrow 4
2	2 \rightarrow 1 \rightarrow 0 2 \rightarrow 1 2 2 \rightarrow 3 2 \rightarrow 3 \rightarrow 4
3	3 \rightarrow 4 \rightarrow 0 3 \rightarrow 2 \rightarrow 1 3 \rightarrow 2 3 3 \rightarrow 4
4	4 \rightarrow 0 4 \rightarrow 0 \rightarrow 1 4 \rightarrow 3 \rightarrow 2 4 \rightarrow 3 4

3.2.2 После отключения маршрутизатора с номером 4

Модель сети: 0 - 1 - 2 - 3

Таблица 4: Таблицы достижимости кольца из 5 маршрутизаторов после отключения маршрутизатора 4

Маршрутизатор	Оптимальные маршруты
0	0 0 → 1 0 → 1 → 2 0 → 1 → 2 → 3
1	1 → 0 1 1 → 2 1 → 2 → 3
2	2 → 1 → 0 2 → 1 2 2 → 3
3	3 → 2 → 1 → 0 3 → 2 → 1 3 → 2 3

3.3 Топология сети звезда

3.3.1 Исходное состояние

Взято 5 маршрутизаторов. Модель сети:

2 - 0

2 - 1

2 - 3

2 - 4

Таблица 5: Таблицы достижимости звезды из 5 маршрутизаторов в исходном состоянии

Маршрутизатор	Оптимальные маршруты
0	0 $0 \rightarrow 2 \rightarrow 1$ $0 \rightarrow 2$ $0 \rightarrow 2 \rightarrow 3$ $0 \rightarrow 2 \rightarrow 4$
1	$1 \rightarrow 2 \rightarrow 0$ 1 $1 \rightarrow 2$ $1 \rightarrow 2 \rightarrow 3$ $1 \rightarrow 2 \rightarrow 4$
2	$2 \rightarrow 0$ $2 \rightarrow 1$ 2 $2 \rightarrow 3$ $2 \rightarrow 4$
3	$3 \rightarrow 2 \rightarrow 0$ $3 \rightarrow 2 \rightarrow 1$ $3 \rightarrow 2$ 3 $3 \rightarrow 2 \rightarrow 4$
4	$4 \rightarrow 2 \rightarrow 0$ $4 \rightarrow 2 \rightarrow 1$ $4 \rightarrow 2$ $4 \rightarrow 2 \rightarrow 3$ 4

3.3.2 После отключения маршрутизатора с номером 4

Модель сети:

2 - 0

2 - 1

2 - 3

Таблица 6: Таблицы достижимости звезды из 5 маршрутизаторов после отключения маршрутизатора 4

Маршрутизатор	Оптимальные маршруты
0	0 $0 \rightarrow 2 \rightarrow 1$ $0 \rightarrow 2$ $0 \rightarrow 2 \rightarrow 3$
1	$1 \rightarrow 2 \rightarrow 0$ 1 $1 \rightarrow 2$ $1 \rightarrow 2 \rightarrow 3$
2	$2 \rightarrow 0$ $2 \rightarrow 1$ 2 $2 \rightarrow 3$
3	$3 \rightarrow 2 \rightarrow 0$ $3 \rightarrow 2 \rightarrow 1$ $3 \rightarrow 2$ 3

4 Выводы

В рамках работы была реализована программная модель, позволяющая объединять произвольное количество роутеров в сети с помощью протокола Open Shortest Path First. Корректность работы алгоритма построения кратчайших путей подтверждена тестированием на трёх видах топологий сети.

По результатам сравнения свойств различных топологий, в предположении равенства весов всех каналов связи, можно сделать следующие выводы:

- При линейной топологии максимальная длина пути между узлами может достигать $n - 1$, где n – число роутеров в сети. При этом выход из строя любого узла, кроме крайних, нарушает связность сети, и разбивает её на две компоненты.
- При звездной топологии максимальная длина пути при любом числе узлов не превосходит 2. Выход из строя любого нецентрального узла не нарушает связность, но при отключении центрального роутера сеть полностью «разваливается», и все остальные узлы оказываются изолированными
- При кольцевой топологии максимальная длина пути в 2 раза меньше, чем в линейной, и может быть порядка $n/2$. При этом в отличие от линейной топологии, выход из строя одного любого узла не нарушает связность, но при выходе из строя двух узлов сеть также распадается на две компоненты.

Также в приложениях приведены протоколы выполнения программы.

5 Ссылка на github

https://github.com/DanilPestryakov/comps_networks

6 Приложения

```
shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 1
(0, 2): 0 -> 1 -> 2
(0, 3): 0 -> 1 -> 2 -> 3
(0, 4): 0 -> 1 -> 2 -> 3 -> 4
shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): 1 -> 2 -> 3 -> 4

shortest ways from 2:
(2, 0): 2 -> 1 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): 2 -> 3 -> 4

shortest ways from 3:
(3, 0): 3 -> 2 -> 1 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): 3 -> 4
shortest ways from 4:
(4, 0): 4 -> 3 -> 2 -> 1 -> 0
(4, 1): 4 -> 3 -> 2 -> 1
(4, 2): 4 -> 3 -> 2
(4, 3): 4 -> 3
(4, 4): 4
```

Рис. 1: Линейная топология. Исходное состояние

```
router0 got: "LSU 2 dropped"

shortest ways from 3:
(3, 0): None
(3, 1): None
(3, 2): None
(3, 3): 3
(3, 4): 3 -> 4
shortest ways from 4:
(4, 0): None
(4, 1): None
(4, 2): None
(4, 3): 4 -> 3
(4, 4): 4

shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): None
(1, 3): None
(1, 4): None

shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 1
(0, 2): None
(0, 3): None
(0, 4): None
```

Рис. 2: Линейная топология. Удаление роутера с номером 2

```
shortest ways from 4:
(4, 0): 4 -> 0
(4, 1): 4 -> 0 -> 1
(4, 2): 4 -> 3 -> 2
(4, 3): 4 -> 3
(4, 4): 4

shortest ways from 3:
(3, 0): 3 -> 4 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): 3 -> 4

shortest ways from 2:
(2, 0): 2 -> 1 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): 2 -> 3 -> 4

shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): 1 -> 0 -> 4

shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 1
(0, 2): 0 -> 1 -> 2
(0, 3): 0 -> 4 -> 3
(0, 4): 0 -> 4
```

Рис. 3: Топология кольцо. Исходное состояние

```
router2 got: "LSU 4 dropped"

shortest ways from 1:
(1, 0): 1 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): None
shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 1
(0, 2): 0 -> 1 -> 2
(0, 3): 0 -> 1 -> 2 -> 3
(0, 4): None

shortest ways from 3:
(3, 0): 3 -> 2 -> 1 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): None

shortest ways from 2:
(2, 0): 2 -> 1 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): None
```

Рис. 4: Топология кольцо. Удаление роутера с номером 4

```
shortest ways from 3:
(3, 0): 3 -> 2 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): 3 -> 2 -> 4

shortest ways from 1:
(1, 0): 1 -> 2 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): 1 -> 2 -> 4

shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 2 -> 1
(0, 2): 0 -> 2
(0, 3): 0 -> 2 -> 3
(0, 4): 0 -> 2 -> 4

shortest ways from 4:
(4, 0): 4 -> 2 -> 0
(4, 1): 4 -> 2 -> 1
(4, 2): 4 -> 2
(4, 3): 4 -> 2 -> 3
(4, 4): 4

shortest ways from 2:
(2, 0): 2 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): 2 -> 4
```

Рис. 5: Топология звезда. Исходное состояние

```

shortest ways from 3:
(3, 0): 3 -> 2 -> 0
(3, 1): 3 -> 2 -> 1
(3, 2): 3 -> 2
(3, 3): 3
(3, 4): None

router2 got: "LSU 0->2"

router2 got: "LSU 1->2"

shortest ways from 0:
(0, 0): 0
(0, 1): 0 -> 2 -> 1
(0, 2): 0 -> 2
(0, 3): 0 -> 2 -> 3
(0, 4): None

router2 got: "LSU 3->2"

shortest ways from 1:
(1, 0): 1 -> 2 -> 0
(1, 1): 1
(1, 2): 1 -> 2
(1, 3): 1 -> 2 -> 3
(1, 4): None

shortest ways from 2:
(2, 0): 2 -> 0
(2, 1): 2 -> 1
(2, 2): 2
(2, 3): 2 -> 3
(2, 4): None

```

Рис. 6: Топология звезда. Удаление роутера с номером 4