

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ
ПЕТРА ВЕЛИКОГО

ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ И МЕХАНИКИ
ВЫСШАЯ ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ФИЗИКИ

**Реализация протокола динамической маршрутизации
Open Shortest Path First
Отчёт по лабораторной работе №2**

Выполнил:

Студент: Ли Жуйци

Группа: 5040102/10201

Принял:

к. ф.-м. н., доцент

Баженов Александр Николаевич

2022 г.

Содержание

1. Постановка задача	2
2. Теория	3
3. Реализация	4
4. Результаты	5
4.1. Линейная топология	5
4.2. Топология «кольцо»	5
4.3. Топология «звезда»	6
5. Обсуждение	7
6. Список литературы	8

1. Постановка задача

Требуется разработать систему из неограниченного количества взаимодействующих друг с другом маршрутизаторов, которые организуются в сеть и обеспечивают передачу сообщений от каждого маршрутизатора к каждому по кратчайшему пути.

Необходимо рассмотреть три топологии сети: линейная, кольцо, звезда. Также необходимо рассмотреть перестройку таблиц достижимости при стохастических разрывах связи.

2. Теория

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

Принцип работы протокола 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), который рассчитывает оптимальный маршрут к каждой сети.

Также выбирается выделенный маршрутизатор (designated router, DR), который управляет процессом рассылки LSA 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. Реализация

Данная работа реализована на языке программирования Python 3.8. Отчёт подготовлен в онлайн-редакторе LaTeX "Overleaf".

Программа разделена на модули:

- **Message**

- **Topology** - Здесь реализован алгоритм Дейкстры.

- **Connection**

- **Router**

Здесь реализован класс маршрутизаторов и выделенного маршрутизатора.

- **Network** - Маршрутизаторы связаны с помощью орграфа с единичными весами рёбер.

- **NEIGHBOURS** (i , $\text{neighbours}(i)$) – сообщение от DR о необходимости добавления новых соседей для узла i ;

- **SET_TOPOLOGY** (topology) – сообщение от DR с информацией о текущей топологии;

- **OFF**(i) – сообщение от DR о необходимости отключения узла i от топологии.

- **NEIGHBOURS** – запрос на добавление в топологию новых соседей;

- **GET_TOPOLOGY** () – запрос на получение от DR информации о текущей топологии;

- **OFF** () – сообщение об отключении маршрутизатора.

4. Результаты

4.1. Линейная топология

Рассмотрим пример работы программы для линейной топологии на примере сети с 5 узлами. Конфигурация этой сети выглядит следующим образом:

Узлы [0, 1, 2, 3, 4]

Связи [[1], [0, 2], [1,3],[2,4],[3]]

К сети подключены все 5 узла. Кратчайшие пути:

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]]

Отключим нулевой узел. Тогда получим следующие кратчайшие пути:

0: [[0], [], [], [], []]

1: [[], [1], [1, 2], [1, 2, 3], [1, 2, 3, 4]]

2: [[], [2, 1], [2], [2, 3], [2, 3, 4]]

3: [[], [3, 2, 1], [3, 2], [3], [3, 4]]

4: [[], [4, 3, 2, 1], [4, 3, 2], [4, 3], [4]]

4.2. Топология «кольцо»

Узлы [0, 1, 2, 3, 4]

Связи [[4, 1], [0, 2], [1, 3], [2, 4], [3, 0]]

К сети подключены все 5 узла. Кратчайшие пути:

0: [[0], [0, 1], [0, 1, 2], [0, 4, 3], [0, 4]]

1: [[1, 0], [1], [1, 2], [1, 2, 3], [1, 0, 4]]

2: [[2, 1, 0], [2, 1], [2], [2, 3], [2, 3, 4]]

3: [[3, 4, 0], [3, 2, 1], [3, 2], [3], [3, 4]]

4: [[4, 0], [4, 0, 1], [4, 3, 2], [4, 3], [4]]

Отключим третий узел. Тогда получим следующие кратчайшие пути:

0: [[0], [0, 1], [0, 1, 2], [], []]

1: [[], [1], [1, 2], [], []]

2: [[], [], [2], [], []]

3: [[], [], [], [3], []]

4: [[4, 0], [4, 0, 1], [4, 0, 1, 2], [], [4]]

4.3. Топология «звезда»

Узлы $[0, 1, 2, 3, 4]$

Связи $[[2], [2], [0, 1, 3, 4], [2], [2]]$

К сети подключены все 5 узлов. Кратчайшие пути:

0: $[[0], [0, 2, 1], [0, 2], [0, 2, 3], [0, 2, 4]]$

1: $[[1, 2, 0], [1], [1, 2], [1, 2, 3], [1, 2, 4]]$

2: $[[2, 0], [2, 1], [2], [2, 3], [2, 4]]$

3: $[[3, 2, 0], [3, 2, 1], [3, 2], [3], [3, 2, 4]]$

4: $[[4, 2, 0], [4, 2, 1], [4, 2], [4, 2, 3], [4]]$

Отключим второй узел. Тогда получим следующие кратчайшие пути:

0: $[[0], [], [], [], []]$

1: $[[], [1], [], [], []]$

2: $[[], [], [2], [], []]$

3: $[[], [], [], [3], []]$

4: $[[], [], [], [], [4]]$

5. Обсуждение

Был реализован и протестирован на различных топологиях протокол OSPF. Данная программа была проверена на трех топологиях, из чего был сделан вывод о ее корректной работе на топологиях: линейная, кольцо, звезда. На основе тестов можно утверждать о работоспособности системы на различных топологиях.

6. Список литературы

1. OSPF [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org/wiki/OSPF>(20.02.2022).
2. <https://github.com/Li-Rui-QI/-CompNetworks/tree/main/Lab2>